

# Desenvolupament d'una aplicació web per a la monitorització de sensors Arduino

---

CaminsSmartLab

Autor: Albert Terrones Muñoz

Director: Carles Farré Tost

Especialitat en Enginyeria del Software

18 d'abril del 2016

Facultat d'Informàtica de Barcelona – Universitat Politècnica de Catalunya



## 1. Resum

---

Actualment els microcontroladors de disseny lliure són cada vegada més complexos i integren una quantitat de sensors i actuadors que els permeten oferir funcionalitats cada vegada més diverses. El seu baix cost, consum i facilitat d'ús, junt amb el desplegament de xarxes de dades sense fil especialitzades, ha permès fer més accessible aquests dispositius a molts àmbits de la nostra vida quotidiana, des de l'industrial fins al sector públic i educatiu. Imaginem fins on podem arribar en un futur quan sensors i actuadors formin part de la nostra vida quotidiana i qualsevol persona sigui capaç de connectar tot el seu entorn; és el que es coneix com l'*Internet de les coses*.

*CaminsSmartLab* permet gestionar i monitoritzar gran quantitat d'aquests sensors de manera centralitzada. L'objectiu principal del projecte consisteix en facilitar que alumnes de grau de l'Escola de Camins tinguin accés a realitzar, mitjançant microcontroladors lliures i de baix cost, experiments en l'àmbit de la geotècnia, hidràulica i materials, normalment reservats a estudiants de màsters i doctorats degut l'alt cost d'aquests experiments. L'Escola de Camins de la Universitat Politècnica de Catalunya, juntament amb l'associació d'estudiants i docents *CaminsMakers*, cerca amb aquest projecte facilitar l'accés i compartició d'aquests experiments i els seus resultats a tota la comunitat educativa.

### Resumen

Actualmente los microcontroladores de diseño libre son cada vez más complejos e integran gran cantidad de sensores y actuadores que permiten ofrecer funcionalidades cada vez más diversas. Su bajo coste, consumo y facilidad de uso, junto con el despliegue de redes de datos inalámbricas especializadas, ha permitido hacer más accesible estos dispositivos a muchos ámbitos de nuestra vida cotidiana, desde el industrial hasta en el sector público y educativo. Imaginemos hasta dónde podemos llegar en un futuro cuando sensores y actuadores formen parte de nuestras vidas cotidianas y cualquier persona pueda conectar todo su entorno; es lo que se conoce como *Internet de las cosas*.

*CaminsSmartLab* permite gestionar y monitorizar gran cantidad de estos sensores de forma centralizada. El objetivo principal del proyecto consiste en facilitar que los alumnos de grado de la Escuela de Caminos tengan acceso a realizar, mediante microcontroladores libres y bajo coste, experimentos en el ámbito de la geotécnica, hidráulica y materiales normalmente reservados a estudiantes de masters y doctorados debido al alto coste de estos experimentos. La Escuela de Caminos de la Universitat Politècnica de Catalunya junto con la asociación de estudiantes y docentes *CaminsMakers* busca con este proyecto facilitar el acceso y compartición de estos experimentos y sus resultados a toda la comunidad educativa.

## *Abstract*

Nowadays open source microcontrollers are becoming more complex and do integrate large number of sensors and actuators that can provide greater functionalities. Its low cost, power consumption and ease of use, along with the deployment of specialized wireless data networks, has enabled these devices to become more accessible to many areas of our daily life, from industrial to the public and educational sector. Imagine what we can achieve when sensors and actuators became part of our daily lives and anyone can connect its environment; It is what is known as the *Internet of things*.

*CaminsSmartLab* allows to manage and monitor large number of these sensors. The aim of the project is to assist students in grade in Civil Engineering with open source and low cost microcontrollers, to have access to experiments in the field of geotechnics, hydraulics and materials that are normally reserved for students of masters and doctorates due to high cost of these experiments. Civil Engineering School of the Polytechnic University of Catalonia in association with students and teachers from *CaminsMakers* seek to ease access and sharing of these experiments and their results to the whole school community.

# Índex

---

<b>1. Resum .....</b>	<b>1</b>
Resumen .....	1
Abstract .....	2
<b>2. Introducció i formulació del problema .....</b>	<b>6</b>
2.1 Laboratoris de l'Escola de Camins de la UPC .....	6
2.2 Què és l'Internet de les coses? .....	8
2.3 Què és Arduino? .....	8
2.4 Motivació personal .....	9
2.5 Objectius del projecte .....	10
<b>3. Estudi previ i gestió del projecte .....</b>	<b>11</b>
3.1 Estat de l'art .....	11
3.2 Anàlisi dels sistemes actuals i conclusions finals .....	12
3.3 Abast del projecte .....	13
3.4 Planificació temporal inicial .....	28
3.5 Identificació i estimació de costos inicial .....	31
<b>4. Anàlisi de requisits .....</b>	<b>36</b>
4.1 Requisits funcionals .....	36
4.2 Requisits no funcionals .....	56
4.3 Model conceptual de dades .....	58
<b>5. Disseny final de la interfície .....</b>	<b>60</b>
5.1 Pàgina principal (Benvinguda i Inici de sessió) .....	60
5.2 Gestió de Laboratoris .....	61
5.3 Gestió d'Experiments d'un Laboratori .....	62
5.4 Gestió de Plantilles de Dispositiu d'un Laboratori .....	62
5.5 Gestió de Dispositius d'un Laboratori .....	62
<b>6. Arquitectura i implementació de <i>CaminsSmartLab</i> .....</b>	<b>63</b>
6.1 Bundle .....	64
6.2 Arquitectura MVC .....	65

6.3	Gestió de la seguretat .....	66
6.4	Entitats i Documents .....	69
6.5	Base de dades relacional .....	71
6.6	Base de dades no relacional .....	71
6.7	Serveis .....	74
6.8	Controladors .....	75
6.9	Capa de Vista.....	77
6.10	Diagrames de Seqüència .....	78
6.11	Descripció de la RESTful API .....	83
6.12	Patrons de disseny .....	84
6.13	Tests .....	87
<b>7.</b>	<b>Tecnologies emprades a la implementació .....</b>	<b>91</b>
7.1	Symfony .....	91
7.2	Eines de DevOps .....	91
7.3	Servidors .....	92
7.4	Eines de persistència i abstracció de base de dades .....	93
7.5	Control de versions GIT.....	95
<b>8.</b>	<b>Planificació, anàlisi de costos i sostenibilitat finals.....</b>	<b>96</b>
8.1	Modificacions realitzades a la planificació .....	96
8.2	Anàlisi econòmic final .....	102
8.3	Sostenibilitat .....	106
<b>9.</b>	<b>Millores pel futur .....</b>	<b>107</b>
<b>10.</b>	<b>Conclusions.....</b>	<b>108</b>
10.1	Aprenent Symfony i l'arquitectura MVC.....	108
10.2	Aprenent DevOps .....	108
10.3	Aplicant la metodologia SCRUM .....	109
<b>11.</b>	<b>Referències .....</b>	<b>110</b>
<b>12.</b>	<b>Bibliografia .....</b>	<b>112</b>

<b>Glossari .....</b>	<b>113</b>
<b>Annex 1 – Diagrama de Gantt a la Planificació Inicial.....</b>	<b>116</b>
<b>Annex 2 – Diagrama de Gantt real .....</b>	<b>118</b>

## 2. Introducció i formulació del problema

---

L'aparició els darrers anys de plataformes *hardware* de disseny lliure<sup>[1]</sup> i baix cost dissenyades per facilitar l'ús i aprenentatge de l'electrònica en múltiples sectors, i més recentment, de xarxes de dades sense fils orientades a les comunicacions d'aquestes plataformes, està ajudant a definir un nou concepte: l'Internet de les coses o *IoT* (*Internet of Things*). Aquest conjunt de tecnologies permet transformar elements quotidians en dispositius interconnectats amb capacitat d'informar del seu estat a una ubicació remota. És així amb les *Smart Cities*<sup>[2]</sup>, l'aplicació de l'*IoT* a les ciutats per tal de millorar els sistemes d'informació de les administracions i de retruc fent un ús més eficient dels equipaments urbans.

En l'àmbit universitari, i pel que fa aquest projecte, els laboratoris de l'Escola de Camins de la UPC, també s'ha volgut aprofitar aquest concepte per millorar els experiments que es realitzen a aquesta escola.

Aquest projecte sorgeix del propòsit per part de l'Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona<sup>1</sup> (ETSECCPB) de millorar el procés d'obtenció d'informació de sensors per a la realització d'experiments docents als laboratoris d'aquesta i facilitar-ne la monitorització i enregistrament d'aquestes dades per l'anàlisi i estudi posteriors en l'àmbit docent.

Actualment aquests experiments generen un gran volum d'informació que s'emmagatzema manualment en fulls de càlcul i no s'utilitza cap eina per organitzar-la ni compartir-la de manera controlada. S'utilitzen eines al «núvol» per compartir aquests fulls de càlcul amb altre personal involucrat, però se'n perd el control. No es té cap eina que permeti monitoritzar sensors en temps real i des de qualsevol lloc. La persona responsable d'un experiment ha d'estar doncs, pendent d'observar els sensors periòdicament per veure l'estat de l'experiment.

Es desitja fer ús d'aquestes tecnologies lliures i de baix cost per millorar l'obtenció de dades d'experiments i centralitzar-les per disposar d'un sistema d'informació comú que faciliti la consulta i estudi als membres d'aquesta comunitat educativa.

### 2.1 Laboratoris de l'Escola de Camins de la UPC

Els laboratoris de l'Escola de Camins<sup>2</sup> són centres de recerca i docència agrupats per departaments de diferents àmbits<sup>[3]</sup> de l'enginyeria civil, obres públiques i geologia. Concretament s'ha treballat als laboratoris de Tecnologia d'Estructures, Geotècnia, Materials i Hidràulica d'aquesta escola.

---

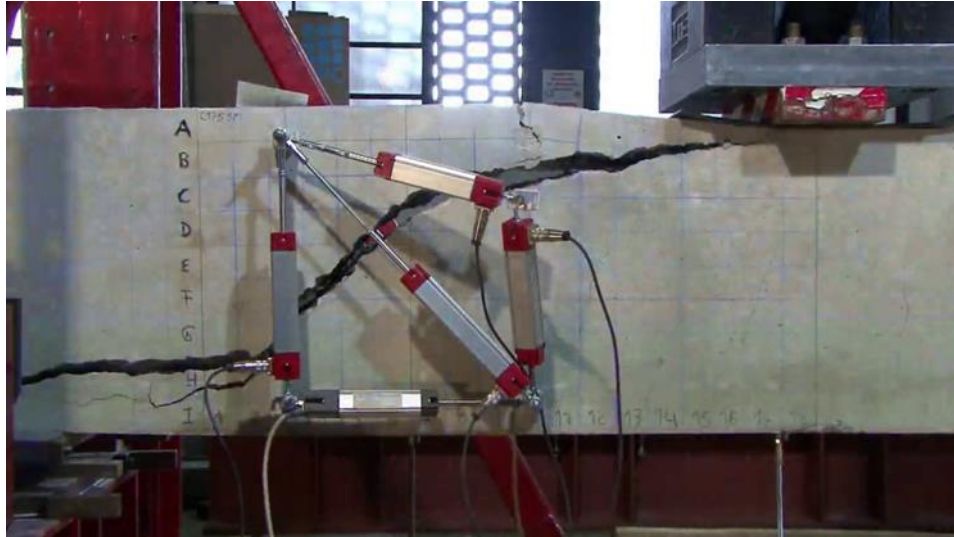
<sup>1</sup> <http://www.camins.upc.edu/>

<sup>2</sup> <https://www.camins.upc.edu/recerca/departaments-centres-i-laboratoris-de-recerca>



A la primera versió del projecte es desitja enregistrar experiments amb sensors de temperatura, humitat, llum, acceleració de dos i tres eixos, pressió, CO2, flexió i galga d'un i tres eixos, però es planteja en un futur ampliar aquesta varietat de sensors.

A dia d'avui, en aquests laboratoris s'hi realitzen experiments docents duts a terme únicament per estudiants de màsters i doctorats, junt amb el personal docent.



*Il·lustració 1: experiment de resistència de material de construcció*



*Il·lustració 2: experiment de tensió-deformació d'una biga*

Aquest projecte sorgeix d'una agrupació d'estudiants i docents de l'Escola de Camins aficionats a les tecnologies anomenada *CaminsMakers* i cerca mitjançant l'ús de dispositius lliures i de baix cost, aconseguir experiments d'un nivell semblant als mencionats, però amb un cost molt menor per fer-los accessibles a la resta d'estudiants i assignatures de grau de l'escola, potenciant així l'aspecte pràctic de les assignatures del grau. Per assolir-ho, s'han especialitzat en l'aprenentatge de programació de

microcontroladors *Arduino*, el funcionament i ús dels quals s'explicarà més endavant. El desenvolupament d'aquest projecte serà dut a terme per l'equip de treball de *CaminsTech* de l'Escola de Camins.

## 2.2 Què és l'Internet de les coses?

L'Internet de les coses (*IoT*) és una xarxa d'objectes físics integrats amb electrònica, *software*, sensors i connectivitat que els permet recollir i intercanviar dades. L'Internet de les coses permet aquests objectes de ser monitoritzats i controlats remotament a través d'una infraestructura de xarxa[4], creant una integració del món físic real als sistemes informàtics, fet que es tradueix en millor eficiència, precisió i economia.



*Il·lustració 3: l'Internet de les coses pot interconnectar vehicles per intercanviar dades*

S'estima que al 2020 cinquanta mil milions de dispositius[5] estaran connectats a Internet. Roba, vehicles, sensors, electrodomèstics, animals,... tots ells interconnectats mitjançant Internet per millorar la seva gestió i eficiència, produint uns 200MB/any/dispositiu de dades. No hi ha dubte que cal preparar-nos per ser capaços de manejar tot aquest volum d'informació.

## 2.3 Què és Arduino?

*Arduino*<sup>3</sup> és una plataforma hardware de codi obert promoguda per l'*Arduino Foundation* que proporciona una capa d'abstracció per facilitar i fer més accessible el disseny de circuits electrònics amb microcontroladors. El sistema proporciona un conjunt de pins d'E/S que permeten interactuar amb sensors, actuadors o altres mòduls externs.

L'entorn de desenvolupament incorpora un llenguatge semblant al C++ basat en *Processing*[6] que facilita la programació del microcontrolador.

---

<sup>3</sup> <https://www.arduino.cc/>

L'abstracció a l'hora de programar aquests dispositius capacita a qualsevol usuari amb pocs coneixements sobre la programació real dels microcontroladors, per programar-los i fer funcionalitats bàsiques de manera més fàcil, com la lectura de sensors o actuadors.

## 2.4 Motivació personal

El maig del 2015, mentre realitzava una beca a la unitat multimèdia de *CaminsTech*<sup>4</sup> i degut a l'auge del grup de treball format per estudiants i professors de l'Escola de Camins anomenat *CaminsMakers*<sup>5</sup>, se'm va proposar de col·laborar amb ells per crear una plataforma que els permetés centralitzar i gestionar de manera eficient un volum de dades considerables que preveïen obtenir mitjançant experiments realitzats als laboratoris pels mateixos estudiants.

El projecte em va semblar molt interessant, i degut a la meva afició per les aplicacions web i l'*Arduino*, i coneixent el bon entorn de treball en el que em trobava em van fer acceptar la proposta.

Tenia coneixements previs en *Symfony*, el *framework* web emprat en aquest projecte, a nivell individual; no obstant això, treballar en equip a *CaminsTech* m'ha fet aprendre molt en tècniques de programació i guanyar experiència en les eines i metodologies de treball actuals.

Finalment, veure el grup d'estudiants de *CaminsMakers* aprenent a treballar en un àmbit en el qual no estan especialitzats (*Arduino* i electrònica en general) em va motivar per ajudar-los en el que pogués, alhora que jo també he après d'ells.

---

<sup>4</sup> <https://caminstech.upc.edu/>

<sup>5</sup> <http://caminsmakers.upc.edu/>

## 2.5 Objectius del projecte

Els objectius principals del projecte són:

### 1. Creació d'una aplicació web per centralitzar dades d'experiments

Crear una aplicació web amb *Symfony* integrada a l'entorn de l'escola amb l'objectiu de permetre a estudiants i personal docent publicar, gestionar i exportar dades d'experiments de manera eficient i alhora centralitzar dades provinents de sensors remots.

### 2. Creació d'una interfície estandarditzada Dispositiu – Servidor

Cal crear una interfície de programació d'aplicació (API) que permeti a dispositius remots (*Arduino*, *Raspberry Pi*, dispositius mòbils, etc.) enviar dades de sensors remots per a emmagatzemar-les i organitzar-les per poder-ne fer la traçabilitat i exportació per al seu estudi posterior.

### 3. Integració de l'aplicació web amb el sistema d'autenticació d'usuaris de la UPC

La gestió d'usuaris serà duta a terme per l'Administrador del sistema, que s'encarregarà de sincronitzar els usuaris del servei de directori de la UPC amb el sistema sota petició dels professors de laboratori, tot assignant diferents rols als usuaris.

No menys important, el meu principal objectiu és saber portar endavant un projecte real, acotat temporalment i d'acord a les expectatives del client, utilitzant les metodologies actuals, i prendre constància de totes les tècniques i coneixements apresos al llarg de la carrera.

### 3. Estudi previ i gestió del projecte

---

#### 3.1 Estat de l'art

S'ha dut a terme un estudi de les eines que existeixen avui dia referent a l'*IoT* que podrien servir per aquest projecte, i malgrat haver-hi molt bones solucions, cap s'adapta exactament a les necessitats de l'Escola de Camins. No obstant això, s'han extret bones idees que han servit per enfocar de manera adient aquest projecte i no “reinventar la roda”.

##### 3.1.1 Sentilo.io

*Sentilo.io*<sup>6</sup> és una plataforma de codi lliure i multiplataforma orientada a les *Smart Cities* i desenvolupada per l'Ajuntament de Barcelona<sup>[7]</sup> que centralitza tot tipus d'informació proporcionada per sensors del mobiliari urbà i infraestructures públiques, i transforma aquestes dades en informació pels sistemes d'informació dels ajuntaments. Les seves principals característiques són:

- Plataforma modular i extensible: està desenvolupat en un llenguatge multiplataforma i de forma modular per a ser fàcilment ampliable i escalable.
- Interfície estandarditzada: utilitza diverses interfícies software estandarditzades per tal de facilitar la comunicació amb diversos models de sensors.
- Definició d'alarmes: permet definir alarmes que es generen en assolir-se uns certs valors definits per l'usuari, canvis d'estat i incomunicació amb el sensor.
- Visualització en temps real: si el sensor ho permet, es permet visualitzar en temps real, o l'última lectura recent del sensor i ubicar-lo sobre un mapa geomètric.
- Gràfiques i exportació: permet generar gràfiques dels sensors i exportar les dades en formats estàndard com fitxers de càlcul aptes per a la importació als sistemes d'informació existents.

Es va idear com un projecte obert i col·laboratiu amb la intenció de treballar amb altres ciutats per tal d'aportar noves idees i funcionalitats. Està publicat sota una llicència LGPL3, fet que permet crear una solució comercial sempre i quan el client final rebi una còpia del codi font d'aquest software.

---

<sup>6</sup> Web del projecte *Sentilo* de l'Ajuntament de Barcelona <http://sentilo.io>

### 3.1.2 TheThings.io

*TheThings.io*<sup>7</sup> és la solució que més s'apropa des del punt de vista tecnològic a la idea del nostre projecte. És una plataforma privada desenvolupada per una empresa barcelonina que s'ofereix com un servei o «SaaS» mitjançant un *backend* i una API flexible. Aquest tipus de servei ens obliga a estar lligats a aquesta empresa en la infraestructura tecnològica. Té un cost anual per sensor i diferents plans de contractació segons les funcionalitats descrites a continuació:

- Interfície estandarditzada: suporta múltiples estàndards oberts per a la comunicació amb els sensors.
- Una interfície web amigable i funcional: proporciona múltiples eines mitjançant una aplicació web intuïtiva que proporciona gràfiques, comptadors de sensors, nombre de peticions/sensor i mapa d'ubicació dels sensors.
- Execució de codi personalitzat a la seva infraestructura: per definir alarmes, permet desenvolupar codi propi i pujar-lo als seus serveis perquè s'executi cada vegada que es processa una dada provinent del sensor i generar una resposta concreta.
- Servei d'anàlisis: permet consultar mètriques per sensors, aplicacions i la seva activitat. Permet obtenir informes personalitzats a la versió «Enterprise».
- Emmagatzematge i crides API: proporciona emmagatzematge limitat en espai de les dades i un límit de crides API per sensor al mes.
- *Open Data*: compartició de les dades i dels sensors amb altres usuaris o mitjançant la publicació via web.
- API per a integració amb aplicacions de tercers: proporciona una interfície web de consulta de les dades per a aplicacions de tercers.

## 3.2 Anàlisi dels sistemes actuals i conclusions finals

S'han analitzat amb profunditat les solucions mencionades i després d'estudiar la viabilitat d'utilitzar cadascuna d'elles s'ha arribat a la conclusió que:

- Es desitja una plataforma allotjada totalment a les infraestructures de la UPC per a la integració amb els recursos d'autenticació dels usuaris d'aquesta universitat.
- Disposar d'una plataforma oberta dins d'una universitat pública afavoreix la democratització i compartició de la informació amb la comunitat educativa.
- No es vol dependre d'empreses externes a la universitat per a aquest servei.

---

<sup>7</sup> Web de l'*startup* TheThings.io: <http://thethings.io>

- Adaptació de la infraestructura als sensors ja existents que no segueixen un format estàndard de comunicació amb les plataformes mencionades a l'estat de l'art.
- Permetre a futurs becaris i estudiants ampliar la plataforma amb noves idees sense dependre de les limitacions d'una plataforma privada externa.

Si bé és cert que el *Sentilo.io* és una solució vàlida ja que compleix aquests criteris en ser software lliure, s'ha conclòs que és una plataforma massa enfocada a les *Smart Cities* i no s'ajusta exactament a les necessitats dels laboratoris de l'escola.

Per aquests motius s'ha decidit implementar una solució a mida per a ús intern de l'Escola de Camins basant-nos en certes funcionalitats de les plataformes estudiades.

### 3.3 Abast del projecte

El projecte es centrarà principalment en assolir l'emmagatzematge de dades d'experiments de laboratoris i la gestió d'aquesta per part dels usuaris de l'escola. L'objectiu és posar en marxa aquest sistema, funcional i documentat per que en un futur pugui ser ampliat i millorat per altres estudiants.

L'abast del projecte serà aconseguir un sistema web on responsables de laboratoris i estudiants de l'Escola de Camins tinguin un repositori web que els permeti gestionar les dades d'experiments i poder exportar-les per analitzar-les posteriorment. Aquesta gestió haurà d'estar supervisada mitjançant diferents rols d'usuari, que tindran accés a diferents funcionalitats dins del sistema.

Existeixen tres tipus de rols d'usuari al nostre sistema:

- Administrador: són responsables de *CaminsTech* que s'encarregaran de gestionar els usuaris i els laboratoris del sistema.
- Responsable d'un laboratori: persona docent assignada a un laboratori. Tindrà permisos per gestionar un laboratori i els seus experiments i dispositius.
- Operador: persona col·laboradora d'un laboratori i encarregada de la publicació de dades als experiments.

A continuació s'exposen els diferents escenaris que contindrà l'aplicació pels diferents rols.

#### 3.3.1 Escenari 1: Sincronització d'usuaris

El nostre sistema proporcionarà una eina CLI desenvolupada amb *Symfony* per realitzar altes i baixes d'usuari manualment o automàticament mitjançant *scripts*. A l'alta s'especificarà el nom d'usuari, contrasenya, correu electrònic, nom públic i rol. No s'ha considerat fer aquesta funcionalitat via web per comoditat de l'administrador.



### 3.3.2 Escenari 2: Pàgina principal (Benvinguda i Inici de sessió)

Aquesta és la pàgina principal que es presentarà a l'usuari a l'entrar. Contindrà la presentació del sistema *CaminsSmartLab* amb el logotip del producte. Els elements a mostrar són:

- Imatge promocional del *CaminsSmartLab* i logotips de *CaminsTech*.
- Formulari d'inici de sessió sol·licitant les credencials de la UPC per accedir al sistema web.

CaminsSmartLab

Log in

## CAS Authentication - UPC

Username

Password

Log In

*Il·lustració 4: mockup de la pàgina principal – Inici de sessió*

### 3.3.3 Escenari 3: Pàgina de gestió de Laboratoris (*Laboratories*)

Un cop l'usuari s'ha identificat amb les seves credencials i ha estat autenticat i autoritzat el primer que veurà serà la **gestió de laboratoris** donats d'alta al sistema.

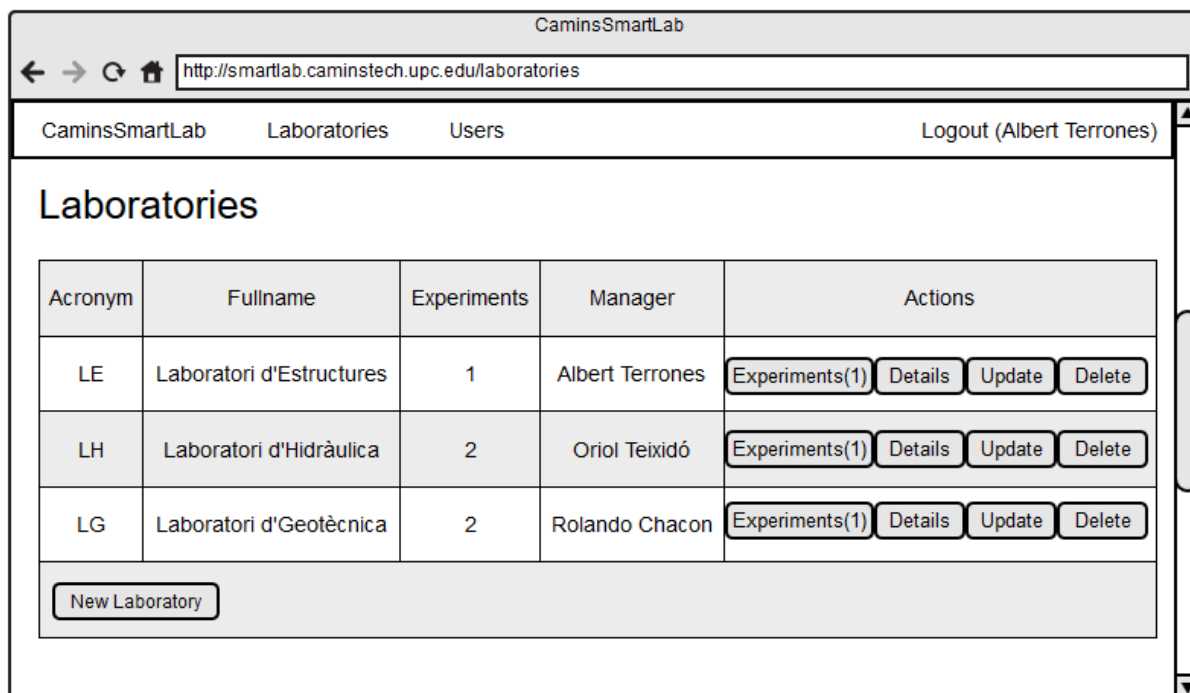


### 3.3.3.1 Llistat de Laboratoris (*List of Laboratories*)

Els Laboratoris són objectes purament informatius amb dades d'interès general, com adreça, especialitat, descripció i persona de contacte d'aquest. Els Laboratoris contenen Experiments, Plantilles de Dispositiu i Dispositius.

Per cada Laboratori es mostrarà:

- *Acronym*: codi a nivell d'UPC per referenciar el laboratori. (Ex. LE – Laboratori d'Estructures).
- *Fullname*: nom complet del laboratori.
- *Experiments*: nombre d'experiments actius que conté el laboratori.
- *Manager*: persona responsable del laboratori amb permisos per gestionar aquell laboratori.
- *Actions*: conjunt d'operacions disponibles per realitzar sobre aquell laboratori; aquestes opcions varien segons el rol de l'usuari que l'està consultant.



Acronym	Fullname	Experiments	Manager	Actions
LE	Laboratori d'Estructures	1	Albert Terrones	Experiments(1) Details Update Delete
LH	Laboratori d'Hidràulica	2	Oriol Teixidó	Experiments(1) Details Update Delete
LG	Laboratori d'Geotècnica	2	Rolando Chacon	Experiments(1) Details Update Delete

New Laboratory

Il·lustració 5: *l·listat de Laboratoris (vista Administrador)*

La vista pels Responsables de Laboratori (*Manager*) serà idèntica però amb accions exclusivament sobre els seus propis laboratoris i sense acció de Nou Laboratori (*New Laboratory*). Tots els laboratoris són accessibles per tots els usuaris a nivell de consulta, independentment del seu rol i associació de *Manager* que tinguï.

La vista pels Operadors d'un laboratori (*Operator*) serà idèntica però amb accions de Detalls i Experiments dels laboratoris i sense acció de Nou Laboratori (*New Laboratory*).

### 3.3.3.2 Creació Edició, Consulta i Eliminació de Laboratoris

Els únics usuaris amb permisos per donar d'alta i baixa Laboratoris seran aquells amb rol d'Administrador. A l'alta l'Administrador haurà d'especificar:

- *Laboratory fullname*: nom complet públic del Laboratori.
- *Laboratory acronym*: sigles d'identificació del laboratori a nivell intern de la UPC.
- *Specialization*: especialitat del departament al que pertany aquell laboratori.
- *Physical Address*: adreça del laboratori.
- *Manager*: persona responsable del laboratori, encarregada de gestionar els experiments i dispositius d'aquell laboratori.
- *Operators*: conjunt d'usuaris que tindran capacitat de publicar i consultar dispositius del laboratori.
- *Short description*: informació de caire general sobre el laboratori.

The screenshot shows a web browser window titled 'CaminsSmartLab' with the URL 'http://smartlab.caminstech.upc.edu/laboratories/create'. The page has a navigation bar with links: 'CaminsSmartLab', 'Laboratories', 'Users', 'Help', and 'Logout (Albert Terrones)'. The main content area is titled 'Laboratories / Create' and contains a form with the following fields:

- Laboratory Fullname**: Text input with placeholder 'Eg. Laboratori d'Hidràulica'.
- Laboratory Acronym**: Text input with placeholder 'Eg. LH'.
- Specialization**: Text input with placeholder 'Eg. Hidràulica'.
- Address**: Text input with placeholder 'Eg. Campus Nord - Edifici D1'.
- Manager**: Dropdown menu with 'Albert Terrones (albert.terrones)' selected.
- Operators**: Dropdown menu with 'Alumne (nom.usuari)' selected.
- Short Description**: Text area with placeholder 'Laboratori d'Hidràulica especialitzat en experiments sobre mecànica de fluids.'

A 'Create' button is located at the bottom of the form.

Il·lustració 6: creació d'un Laboratori

La vista d'edició (*Update*) serà idèntica, només amb canvis visuals de “*Create*” per “*Update*” i la vista de Detalls (*Details*) sense botó i amb els camps de només lectura.

### 3.3.4 Escenari 4: Pàgina de gestió d'Experiments (*Experiments*)

Aquesta pàgina permet la gestió dels Experiments dins d'un Laboratori concret. Depenent del seu rol, l'usuari podrà crear Experiments (*Experiment*) o només visualitzar-lo. Un Experiment pot estar obert o tancat a noves dades de dispositius.

#### 3.3.4.1 Llistat d'Experiments d'un Laboratori (*List of Experiments of a Laboratory*)

La pàgina principal de gestió d'experiments conté un llistat de tots els experiments d'un laboratori. D'aquests se'n desprèn la següent informació:

- *Code*: codi intern del laboratori per referenciar un experiment.
- *Full Name*: nom explicatiu de l'experiment.
- *Last Data received*: indica la data de l'última dada rebuda per part dels dispositius d'aquell Experiment.
- *Open?*: indica si un experiment està habilitat per rebre noves dades de dispositius.
- *Actions*: consultar els seus Dispositius (*Devices*), detalls sobre l'Experiment, actualització, *DeviceTemplate* i eliminació.

CaminsSmartLab

[←](#) [→](#) [↺](#) [🏠](#) <http://smartlab.caminstech.upc.edu/laboratories/1/experiments>

CaminsSmartLab   Laboratories   Users   [Logout \(Albert Terrones\)](#)

Laboratories / LE / Experiments

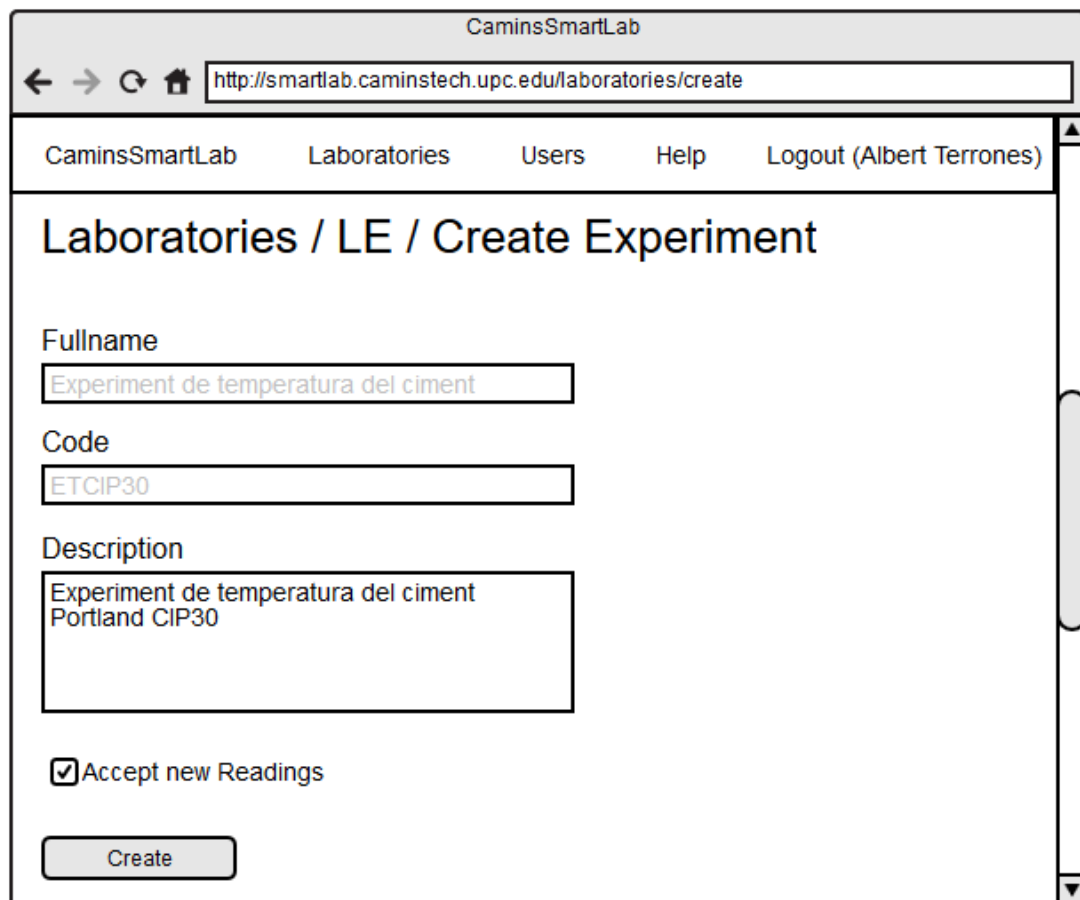
Code	Fullname	Last Data received	Open?	Actions
ETCIP30	Temperatura del Ciment CP30	25/11/2015	Yes	<a href="#">Devices(1)</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
EHLAB1	Humitat Lab 1 Ciment	25/11/2015	Yes	<a href="#">Devices(1)</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
ELAB1	Lluminositat	25/11/2015	No	<a href="#">Devices(1)</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Experiment](#)

*Il·lustració 7: Llistat d'experiments del laboratori l.e (vista Manager)*

### 3.3.4.2 Creació, Edició, Consulta i Eliminació d'Experiments d'un Laboratori

Aquesta acció podrà ser duta a terme per part del responsable d'aquell laboratori (*Manager*) i pels Administradors de sistema.

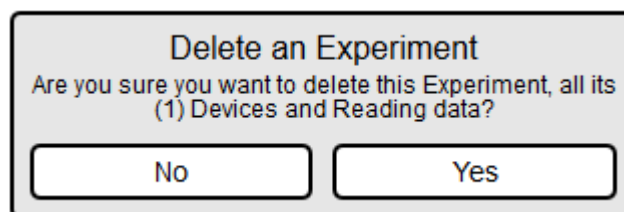


The screenshot shows a web browser window titled 'CaminsSmartLab'. The address bar displays 'http://smartlab.caminstech.upc.edu/laboratories/create'. The navigation menu includes 'CaminsSmartLab', 'Laboratories', 'Users', 'Help', and 'Logout (Albert Terrones)'. The main heading is 'Laboratories / LE / Create Experiment'. The form contains the following fields:

- Fullname:** A text input field containing 'Experiment de temperatura del ciment'.
- Code:** A text input field containing 'ETCIP30'.
- Description:** A text area containing 'Experiment de temperatura del ciment Portland CIP30'.
- Accept new Readings:** A checkbox that is checked.
- Create:** A button at the bottom of the form.

*Il·lustració 8: creació d'un Experiment al Laboratori L.E (vista Manager)*

L'eliminació d'un Experiment implicarà l'eliminació en cascada de tots els Dispositius (*Devices*) i Lectures (*ReadingData*) d'aquest experiment. Aquesta acció només pot ser efectuada pel Responsable de laboratori o un Administrador.



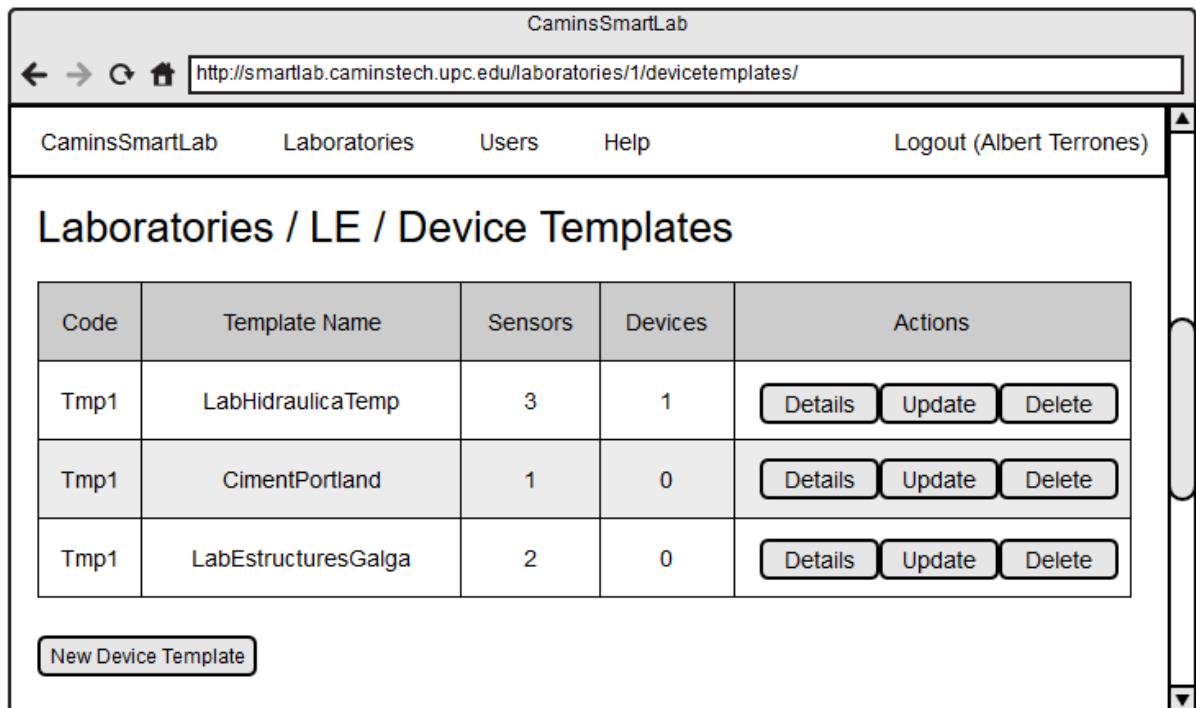
The dialog box is titled 'Delete an Experiment'. It contains the text: 'Are you sure you want to delete this Experiment, all its (1) Devices and Reading data?'. At the bottom, there are two buttons: 'No' and 'Yes'.

*Il·lustració 9: eliminació d'un experiment i totes les dades associades*

### 3.3.5 Escenari 5: Pàgina de gestió de Plantilles de Dispositiu (*Device Template*)

Les plantilles de dispositiu (*Device Template*) són objectes que defineixen l'estructura de dades que envien els dispositius *Arduino* dels laboratoris. El concepte de plantilla permet reaprofitar la configuració d'un dispositiu físic (*Arduino*) per a posteriors experiments.

#### 3.3.5.1 Llistat de Plantilles de Dispositiu d'un Laboratori



CaminsSmartLab

http://smartlab.caminstech.upc.edu/laboratories/1/devicetemplates/

CaminsSmartLab Laboratories Users Help Logout (Albert Terrones)

## Laboratories / LE / Device Templates

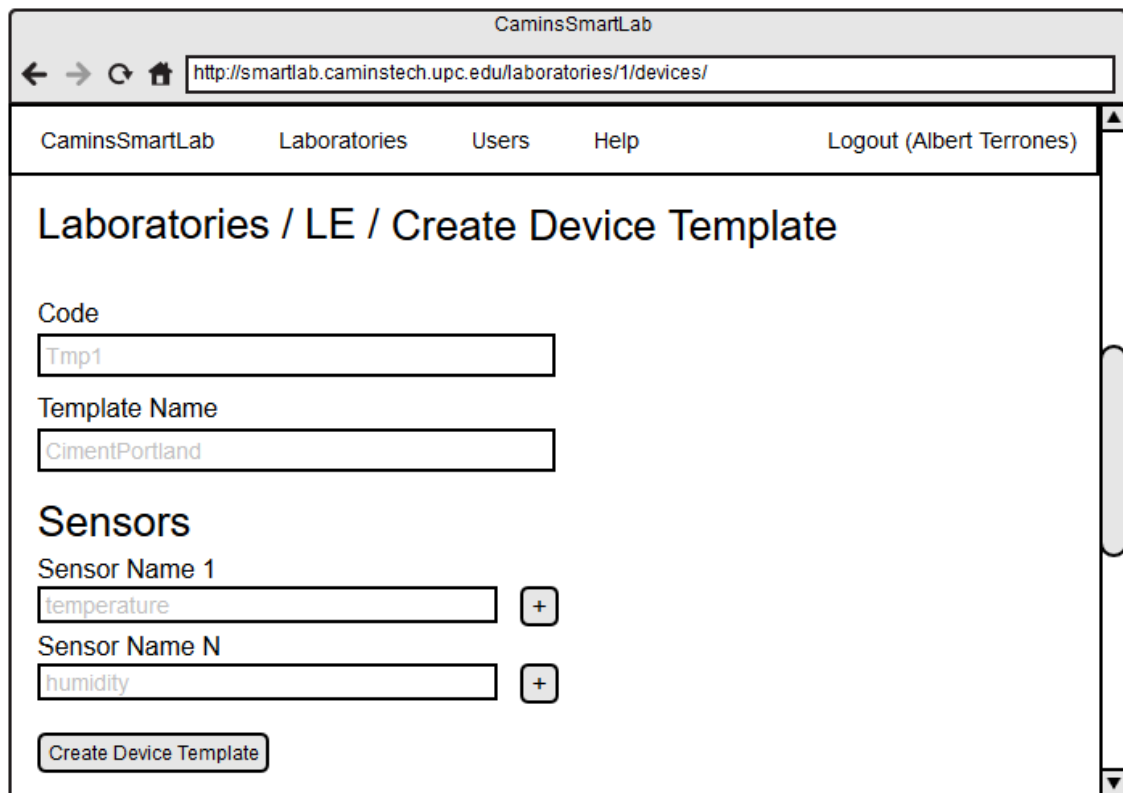
Code	Template Name	Sensors	Devices	Actions
Tmp1	LabHidraulicaTemp	3	1	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
Tmp1	CimentPortland	1	0	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
Tmp1	LabEstructuresGalga	2	0	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Device Template](#)

Il·lustració 10: Llistat de Plantilles de Dispositiu

### 3.3.5.2 Creació, Edició, Consulta i Eliminació de Plantilles de Dispositiu

La creació d'una Plantilla de Dispositiu (*Device Template*) permet reaprofitar els dispositius per a posteriors experiments.



The screenshot shows a web browser window with the address bar displaying `http://smartlab.caminstech.upc.edu/laboratories/1/devices/`. The page title is "CaminsSmartLab". The navigation bar includes links for "Laboratories", "Users", "Help", and a "Logout (Albert Terrones)" button. The main content area is titled "Laboratories / LE / Create Device Template". It contains a form with the following fields: "Code" with the value "Tmp1", "Template Name" with the value "CimentPortland", and a "Sensors" section. The "Sensors" section has two rows: "Sensor Name 1" with the value "temperature" and a "+" button, and "Sensor Name N" with the value "humidity" and a "+" button. At the bottom of the form is a "Create Device Template" button.

*Il·lustració 11: creació d'una Plantilla de Dispositiu*

### 3.3.6 Escenari 6: Pàgina de gestió de Dispositius (*Device*)

Els Dispositius o recursos de dades (*Devices*) són objectes que representen pròpiament els dispositius físics que s'utilitzen en un Laboratori. Cada un d'aquests dispositius està vinculat a un *DeviceTemplate*.

### 3.3.6.1 Llistat de dispositius d'un Experiment (*List of Devices of Experiment*)

CaminsSmartLab

← → ↻ 🏠 <http://smartlab.caminstech.upc.edu/laboratories/1/devices/>

CaminsSmartLab Laboratories Users Help Logout (Albert Terrones)

## Laboratories / LE / ETCIP30 / Devices

Code	FullName	Last reading	Template	Actions
ARD1	Arduino Marc	2015-12-02 15:42:32	CimentPortland	<a href="#">Download</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
ARD2	Arduino Rolando	2015-12-02 15:32:22	CimentPortland	<a href="#">Download</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>
ARD3	Mobile Héctor	2015-12-02 14:42:13	LabHidraulicaTemp	<a href="#">Download</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Device](#)

*Il·lustració 12: Llistat de Dispositius (vista Operador autoritzat)*

### 3.3.6.2 Creació, Edició i Eliminació de dispositius d'un Experiment

CaminsSmartLab

← → ↻ 🏠 <http://smartlab.caminstech.upc.edu/laboratories/1/devices/>

CaminsSmartLab Laboratories Users Help Logout (Albert Terrones)

## Laboratories / LE / ETCIP30 / Create Device

Code

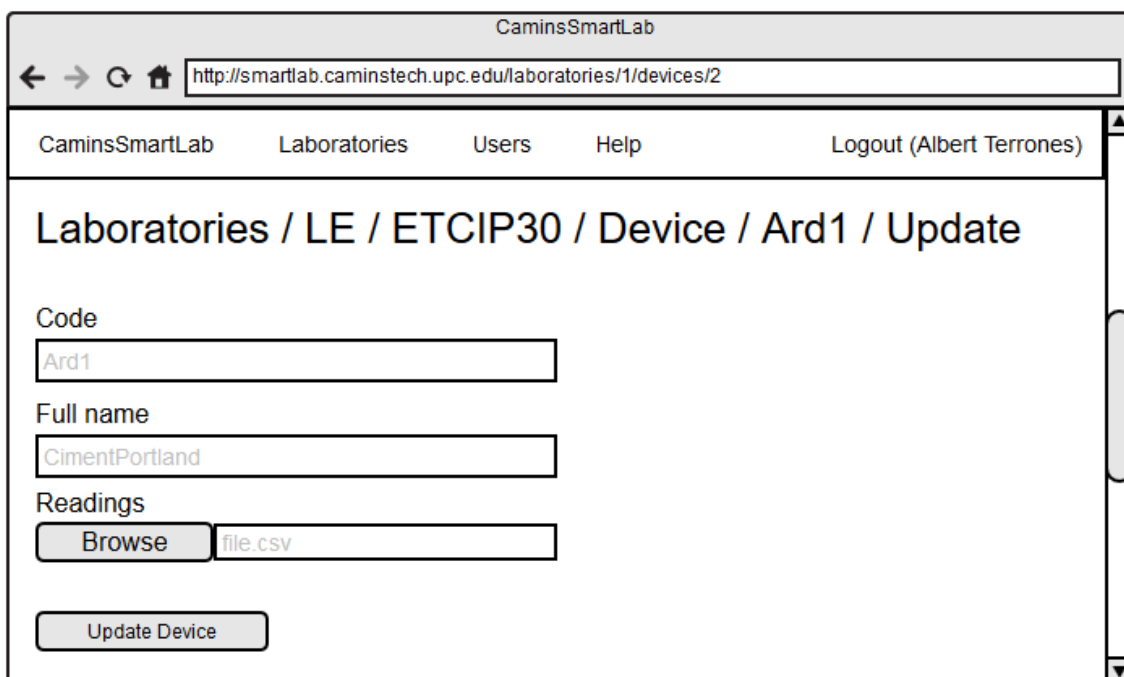
Full name

Template

[Create Device](#)

*Il·lustració 13: creació d'un Dispositiu*

L'actualització del Dispositiu permet, a més, adjuntar un fitxer CSV que inclogui dades de sensors (*Readings*) seguint l'estructura imposat per la Plantilla de Dispositiu.



CaminsSmartLab

← → ↻ 🏠

CaminsSmartLab Laboratories Users Help Logout (Albert Terrones)

## Laboratories / LE / ETCIP30 / Device / Ard1 / Update

Code

Full name

Readings

*Il·lustració 14: edició i/o adjunt de Lectures (Readings) d'un Dispositiu*

### 3.3.6.3 Consulta dels detalls d'un dispositiu i exportació de lectures d'un Experiment

Aquesta és la pàgina més important per un Operador a l'hora de connectar un dispositiu físic amb el sistema web. A aquesta pàgina es mostren dades de caire informatiu del dispositiu, en concret:

- Codi intern del dispositiu: referencia un dispositiu físic equivalent.
- *Fullname*: nom públic del dispositiu.
- *Template*: indica a quina Plantilla de Dispositiu pertany.
- *Create Date Time*: data i hora de creació del dispositiu.
- *Last Data Input*: indica a quina data i hora s'ha rebut l'última dada del dispositiu físic.
- *REST API PUT METHOD*: adreça HTTP del dispositiu virtual per a la publicació de dades de sensors mitjançant la REST API del sistema web.
- *Token*: token privat autogenerat pel sistema per assignar al dispositiu físic.
- Taula de dades: permet consultar les dades rebudes per part del dispositiu.



CaminsSmartLab

← → ↻ 🏠

http://smartlab.caminstech.upc.edu/laboratories/1/devices/3

CaminsSmartLab   Laboratories   Users   Help   Logout (Albert Terrones)

Laboratories / LE / ETCIP30 / Device / Ard1

Code

ARD1

Full name

Arduino Marc

Template

CimentPortland

Create Date Time

2015-12-20 20:05

Last Data Input

2015-12-20 20:26:35

REST API: PUT METHOD

http://smartlab.caminstech.upc.edu/api/devices/3

Token

e465k2313j23h4g5

Filter...

▼ Temperature	▼ Humidity
15	35
16	40

1 | 2 | 3

Export to CSV

Il·lustració 15: detalls d'un dispositiu

### 3.3.7 Eines de desenvolupament

Les tecnologies que emprarem per desenvolupar aquest projecte les podem dividir en dos eixos ben diferenciats:

- Arquitectura servidor web: la tecnologia principal emprada en aquest eix serà ***Symfony2.8 LTS***, un *framework* web basat en el llenguatge PHP orientat al desenvolupament d'aplicacions MVC, altament configurable, modular, escalable i de fàcil manteniment, que inclou funcionalitats comunes com suport multiidioma, seguretat, ACL (*Access Control Lists*) i mecanismes per evitar vulnerabilitats web conegudes.
- Arquitectura client: inicialment es treballarà amb la plataforma *Arduino Yun* que incorpora connectivitat sense fils i suport per a connexions HTTP necessaris per a realitzar peticions a la REST API del sistema. Aquests dispositius disposen de sensors programats pels propis alumnes de l'Escola de Camins que els serveixen per realitzar experiments. En un futur es podrà canviar la plataforma de desenvolupament del client per qualsevol altra sempre que es compleixin els contractes de les operacions establerts a la REST API.

En quant a l'emmagatzematge de dades, s'utilitzaran dues bases de dades de paradigmes molt diferents:

- Per a l'emmagatzematge de totes les dades d'usuaris, laboratoris, experiments i dispositius s'utilitzarà una sistema gestor de base de dades relacional anomenat *MySQL*, que ens permet mantenir l'esquema conceptual plantejat.
- Per a l'emmagatzematge de lectures de sensors de dispositius (*Readings*) s'utilitzarà un sistema gestor de base de dades no-relacional (*NoSQL*) i sense esquema anomenat *MongoDB*. Aquest és més eficient a l'hora d'emmagatzemar grans volums de dades simples que no necessiten estar relacionades amb cap altra dada i per la naturalesa d'aquestes, no requereixen tant de les propietats ACID com de les BASE.

Posteriorment es detallarà amb més concreció com s'ha dut a terme aquesta distribució de les dades.

### 3.3.8 Justificació de les tecnologies

L'ús d'*Arduino* a la part Client del projecte és degut a exigències per part dels responsables de *CaminsMakers*, ja que és la seva especialitat i a més encaixa correctament amb l'eix Servidor de tot el sistema al ser interoperables entre elles mitjançant l'ús d'estàndards web.

L'elecció de *Symfony* com a *framework* de desenvolupament web ve motivada, a part de per les característiques favorables a la seva elecció, per l'experiència en aquest en

altres projectes desenvolupats al llarg de la carrera i sota l'aprovació del cap de *CaminsTech*.

### 3.3.9 Possibles obstacles i plans de contingència

Els esdeveniments que poden alterar el correcte desenvolupament de tot el sistema són:

- **Desviació temporal:** tant el Treball Final de Grau, com la beca de col·laboració amb l'Escola de Camins tenen una data límit fixada. Qualsevol esdeveniment que pugui alterar la planificació temporal s'haurà de gestionar degudament perquè no tingui repercussions majors.
- **Imprevistos amb la tecnologia software i hardware:** malgrat conèixer amb cert nivell les tecnologies emprades, sempre hi ha un grau de desconeixement i incertesa lligat a les noves funcionalitats que poden alterar el ritme de treball estimat.
- **Rendiment del sistema degut al volum d'informació:** degut a que hi ha dispositius *offline* que generen dades amb una freqüència de fins a 200 Hz, depenent de la seva durada poden produir gran quantitat d'informació i provocar que la inserció, extracció i consulta d'aquestes dades sigui molt costosa en temps i recursos informàtics. S'haurà d'estimar la capacitat màxima de sensors *online* i la quantitat d'informació que es pot analitzar en bloc.

Malgrat no hem establert cap restricció en el nombre de peticions a la REST API per unitat de temps, s'ha confirmat que els dispositius *online* generaran com a molt, una lectura per minut (0,016Hz) i sabem que el sistema és perfectament capaç d'acceptar aquest volum de peticions.

### 3.3.10 Metodologia i rigor

La metodologia que s'ha considerat més apropiada<sup>[8]</sup> pel desenvolupament d'aquest projecte és la metodologia àgil *SCRUM*<sup>8</sup>. L'elecció d'aquesta metodologia va lligada a la voluntat de realitzar un seguiment constant de l'evolució del projecte i seguir un desenvolupament alineat amb les expectatives del client.

---

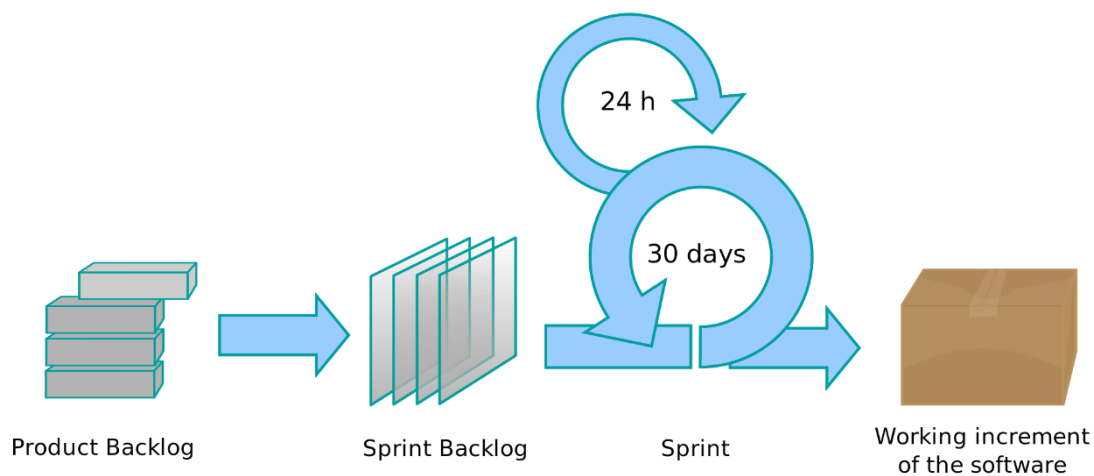
<sup>8</sup> <https://www.scrum.org/>

### 3.3.10.1 Metodologia SCRUM

Com la metodologia *SCRUM* descriu, es proporcionaran entregues periòdicament dels resultats i processos intermedis en períodes de durada fixa (2 setmanes) lliurant els resultats al final de cada iteració. Hi haurà reunions periòdiques amb un responsable de laboratoris i l'equip tècnic de *CaminsTech* per tal de definir més concretament quin serà el transcurs de la propera iteració, així com el diagnòstic de riscos que puguin afectar el calendari establert. L'*SCRUM* estableix que primerament cal definir una llista anomenada *Product Backlog* on s'afegeixen totes les tasques i idees que es considera d'incorporar al projecte. Aquestes tasques estan ordenades per prioritat segons s'acorda entre l'*Scrum Master* i el *Product Owner* a cada *Sprint Review*.

Un *Sprint* defineix un període definit per una unitat de temps que s'estableix a cada equip de treball i permet indicar quin pes té cada tasca acord a la velocitat de l'equip. Els nostres *Sprints* de desenvolupament tindran una durada de 15 dies. Al assolir-se aquest període, es realitzarà un *Sprint Review* on s'analitzarà la feina feta i segons els resultats, l'*Scrum Master* planificarà la propera *Sprint*. L'*Scrum Master* és l'encarregat de trobar solucions a la eventualitat del no assoliment de les fites.

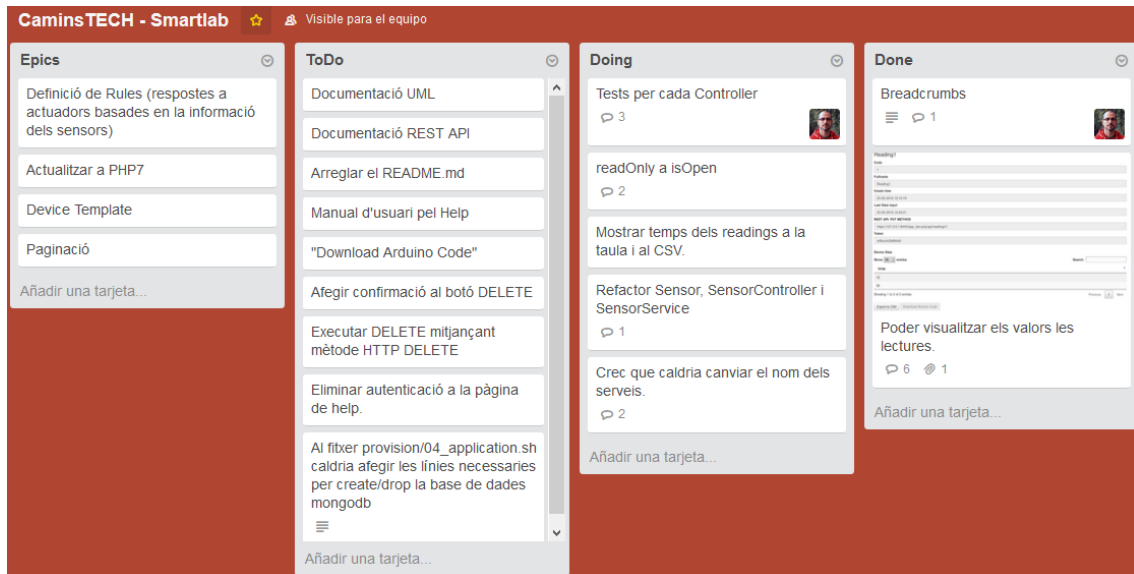
El *Product Owner* o client podrà conèixer l'estat del projecte en tot moment i validar les tasques finalitzades.



*Il·lustració 16: representació gràfica de la metodologia SCRUM amb Sprint a 30 dies*

### 3.3.10.2 Eines de seguiment

Les eines de seguiment que utilitzarem per seguir la metodologia àgil és el servei web *Trello*<sup>9</sup> que ens permetrà definir quatre llistes: *Epics*, *ToDo*, *Doing* i *Done*, que fan referència a la llista d'idees i funcionalitats, llista de tasques seleccionades per fer i ordenades per prioritats, tasques que s'estan realitzant a l'*Sprint* actual i tasques finalitzades. També es farà ús d'eines col·laboratives i de control de versions de software GIT<sup>10</sup>, que ens permetrà tenir un registre de canvis i errors al codi, així com un millor control de versions i de correcció d'incidències.



Il·lustració 17: captura real del Trello emprat a CaminsTech per aquest projecte

### 3.3.10.3 Metodologia de validació

Gràcies a la metodologia àgil, es duren a terme validacions periòdiques per seguir l'evolució del projecte. A l'inici de cada *Sprint* es durà a terme l'*Sprint Review* que permet fer un anàlisi de l'estat d'aquell període i prendre decisions en base als resultats de l'anterior. A cada reunió amb els responsables de laboratoris i l'equip tècnic de *CaminsTech* es presentarà la feina feta en aquella iteració, es comentaran els imprevistos i problemes identificats per tal de tenir una visió global de l'evolució i es permetrà definir quines són les tasques per la següent iteració.

Les reunions amb el director del projecte, Carles Farré Tost, permetran tenir una idea de si el projecte evoluciona favorablement i quins punts cal remarcar per assolir les fites establertes.

<sup>9</sup> Servei web de gestió de tasques <http://trello.com>

<sup>10</sup> Servei web GIT de repositori de codi col·laboratiu <http://bitbucket.org>

### 3.4 Planificació temporal inicial

Després d'haver definit l'abast del projecte, s'ha passat a determinar la següent planificació temporal basada en tasques.

#### 3.4.1 Calendari

El temps estimat per a realitzar el projecte és de 4 mesos exactament. El projecte es va iniciar el dia 1 de setembre del 2015 i es preveu la finalització de la implementació el dia 27 de novembre del mateix any. Es preveu destinar tot el mes de desembre per completar el lliurament final de la Memòria, donant un marge de 7 dies per realitzar correccions de la documentació i perfeccionar la lectura.

La dedicació del temps a la realització del projecte per la meua part serà de 25 hores setmanals en horari laboral de dilluns a divendres de 9:00 a 14:00 h.

#### 3.4.2 Iteracions del desenvolupament

El procés de desenvolupament d'aquest projecte es sotmetrà a una metodologia àgil *SCRUM*. D'aquesta manera el desenvolupament es dividirà en 5 etapes que anomenarem *Sprints* que tindran una durada determinada; cada *Sprint* contindrà un conjunt de tasques més petites que permetran realitzar una estimació dels costos i durada més realistes. Aquesta relació entre *Sprint* i tasques es podrà veure en profunditat al punt [3.4.4 – Diagrama de Gantt](#).

##### 3.4.2.1 *Sprint Inception*: Investigació, planificació i posada en marxa de l'entorn de desenvolupament

Aquest *Sprint* inicial té com a objectiu investigar com es realitzarà el projecte i amb quines eines es farà. Permet preparar l'entorn de desenvolupament i familiaritzar-se amb les tecnologies emprades. S'estudiarà el context i l'abast del projecte, posteriorment es definirà el model de casos d'ús i el model de l'anàlisi a implementar. Finalment, es realitzarà tota la planificació temporal i econòmica del projecte.

El document final d'aquest *Sprint* contindrà el model dels casos d'ús, el model d'anàlisi, *mockups* de les pantalles i documents de la gestió del projecte, els quals seran validats pel *Product Owner* durant l'*Sprint Review*.

Aquesta iteració no té cap dependència de precedència.

#### 3.4.2.2 *Sprint 1*: Implementació Gestió de Laboratoris, Experiments, Plantilles de Dispositiu, Dispositius i Lectures de Dispositiu

En aquest *Sprint* es durà a terme la definició de l'estructura de l'aplicació web i la implementació de les funcionalitats de gestió elementals (alta, baixa, modificació, eliminació i consulta) de laboratoris, experiments, plantilles de dispositiu, dispositius i lectures de sensors. En aquesta iteració però, no s'implementaran el control d'accessos pels usuaris del sistema.

Aquesta iteració té una dependència de precedència que és l'*Sprint Inception* definit al punt [3.4.2.1 – Sprint Inception](#) del document.

#### 3.4.2.3 *Sprint 2*: Implementació Gestió d'Usuaris i REST API

En aquest sprint es durà a terme la implementació de les funcionalitats de la gestió dels usuaris de tipus Administrador, Responsable de Laboratori i Operadors de Laboratori mitjançant la integració amb el servei d'autenticació i autorització de la UPC pel control d'accés a les diferents seccions del sistema web. S'implementaran els següents apartats del sistema web segons el rol de l'usuari en qüestió:

- Els usuaris de tipus Administrador podran accedir a la secció Gestió de Laboratoris i assignar usuaris a laboratoris per fer-los responsables. En aquest apartat es podran observar valors numèrics sobre el nombre d'experiments realitzats per laboratori, nombre d'usuaris i dates d'esdeveniments.
- Els usuaris de tipus Responsable de Laboratori podran accedir a la secció Gestió de Laboratoris per assignar usuaris de tipus Operador al seu laboratori. En aquesta secció podrà donar d'alta nous experiments i sensors i associar-los entre si.
- Els usuaris de tipus Responsable de Laboratori i Operador de Laboratori podran accedir a la secció *Experiments* per enregistrar, consultar i exportar lectures de dispositius.

Finalment s'implementarà una interfície software d'interacció per a aplicacions terceres mitjançant una API que permetrà realitzar accions i consultes a les gestions d'experiments i lectures de sensors, així com l'autenticació d'usuaris.

Aquesta iteració té dependència de precedència amb l'*Sprint 1*.

#### 3.4.2.4 *Sprint 3*: Implementació dels Tests, client HTTP per *Arduino Yun* i configuració del dispositiu en xarxa

En aquest *Sprint* es duran a terme els Tests funcionals per validar la correctesa de les funcionalitats dels controladors, així com l'estudi i implementació del codi per a

*l'Arduino Yun*<sup>11</sup> que farà de pont entre la lectura dels sensors del dispositiu i el servidor web mitjançant l'ús de la REST API implementada.

Aquesta iteració té dependència de precedència amb *l'Sprint 2*.

#### 3.4.2.5 *Sprint 4*: Documentació i lectura finals

En aquest *Sprint* final es durà a terme l'elaboració de la Memòria del projecte i la preparació de la lectura del TFG.

Aquesta iteració té dependència de precedència amb *l'Sprint 3*.

#### 3.4.3 Taula de costos

Fases	Calendari	Durada (hores)	Durada (dies 5h/dia)
<b><i>Sprint Inception</i></b>	01/09/2015 – 25/09/2015	95 hores	19 dies
<b><i>Sprint 1</i></b>	28/09/2015 – 16/10/2015	75 hores	15 dies
<b><i>Sprint 2</i></b>	19/10/2015 – 06/11/2015	75 hores	15 dies
<b><i>Sprint 3</i></b>	09/11/2015 – 27/11/2015	75 hores	15 dies
<b><i>Sprint 4</i></b>	30/11/2015 – 31/12/2015	120 hores	24 dies
		<b>440 hores</b>	<b>Total: 88 dies</b>

*Taula 1: fases de desenvolupament àgil*

#### 3.4.4 Diagrama de Gantt inicial

A l'[Annex 1](#) d'aquest document es pot trobar el Diagrama de Gantt inicial dissenyat per assolir les fites marcades fent ús de la metodologia SCRUM i afegit dins del període de 4 mesos establert.

#### 3.4.5 Valoració d'alternatives i pla d'acció

A causa de la naturalesa dels projectes de software en si mateix, la planificació pot patir desviacions positives i negatives a l'estimació de les durades de les iteracions. Si en alguna de les iteracions no s'assoleixen els objectius establerts d'implementació, es presentarà el problema a la següent *Sprint Review* i es decidirà aleshores si cal retallar temps d'una altra iteració per completar-la, o per contra, cal donar més prioritat a la nova iteració i per tant caldrà descartar tasques per tal d'assolir la fita del lliurament final.

Com la implementació s'espera que estigui finalitzada el mes de novembre, i el mes de desembre està destinat a realitzar la documentació final i la preparació de la lectura (*Sprint 4*), si en el pitjor dels casos la implementació no s'ha finalitzat, es reservarà un marge de 4 hores al dia la primera setmana de desembre (20 hores) a acabar de completar la implementació alhora que s'inicia *l'Sprint 4* destinat a la documentació.

Si per el contrari, els *Sprints* es finalitzen abans d'hora, es reservaran aquestes hores per a futurs imprevistos i s'avançaran els *Sprint* següents. En el cas de la finalització

---

<sup>11</sup> <https://www.arduino.cc/en/Main/ArduinoBoardYun>



anticipada de l'Sprint 4, es destinarà el temps restant per aportar millores al producte final.

### 3.5 Identificació i estimació de costos inicial

S'ha realitzat un estudi sobre la viabilitat econòmica del projecte analitzant els costos directes i indirectes associats al desenvolupament d'aquest projecte així com un pla de contingència en cas de desviació econòmica i un fons de reserva per imprevistos. Finalment es proporciona un resum del pressupost total estimat del cost del projecte.

#### 3.5.1 Costos directes per activitat

Els costos directes per activitat inclouen els recursos humans involucrats en cada una de les etapes d'inici a fi del desenvolupament del projecte definides al diagrama de *Gantt*. Cal avisar però, que durant el TFG sóc jo qui realitza totes les activitats associades als rols que es descriuen a continuació, però per realitzar l'estudi de viabilitat econòmica s'ha volgut tenir una "visió d'empresa" on cada activitat correspondria a un rol específic.

La llista dels perfils necessaris i la seva remuneració anual és la següent:

Perfil	Remuneració anual (€)	Remuneració (€/h)
Cap de projecte	35.000 €	18,23 €/h
Analista funcional	33.000 €	17,19 €/h
Dissenyador UX	18.000 €	9,38 €/h
Programador Web	18.000 €	9,38 €/h
Programador Arduino/Python	18.000 €	9,38 €/h
Responsable QA	22.000 €	11,46 €/h

*Taula 2: costos per perfil*

Tot seguit es mostren els costos estimats per activitat i perfil tal com està definit al diagrama de *Gantt* de la planificació inicial:

Activitat	Unitat	Perfil	Preu unitari (€/h)	Cost total (€)
Investigació i contextualització	26 h	Cap de projecte	18,23	473,98
Definició de requisits funcionals	12 h	Analista de requeriments	17,19	206,28
Definició de tasques i full de ruta	22 h	Cap de projecte	18,23	401,06
Prototipatge UX	15h	Dissenyador UX	9,38	140,7
Eines i entorn desenvolupament i Review	20 h	Programador web	9,38	187,6
Sprint 1 – Definició estructura web	11 h	Programador Web	9,38	103,18
Sprint 1- Implementació casos d'ús	52 h	Programador Web	9,38	487,76
Sprint 1 – Unit Testing	10 h	Software Tester	11,46	114,6
Sprint 1 - Review	2 h	Cap de projecte	18,23	36,46
Sprint 2 – Implementació CU Gestió Usuaris	15 h	Programador Web	9,38	140,7
Sprint 2- Implementació CU Rest API + MongoDB	45 h	Programador Web	9,38	422,1
Sprint 2 – Unit Testing	13 h	Software Tester	11,46	148,98
Sprint 2 - Review	2 h	Cap de projecte	18,23	36,46
Sprint 3 – Definició estructura Arduino Yun Client	15 h	Programador Pt	9,38	56,28 140,7
Sprint 3 - Implementació Tests Funcionals	54 h	Programador Pt	9,38	506,52
Sprint 3 – Unit Testing	4 h	Software Tester	11,46	45,84
Sprint 3 - Review	2 h	Cap de projecte	18,23	36,46
Sprint 4 – Redacció de la memòria	60 h	Cap de projecte	18,23	1093,8
Sprint 4 – Preparació de la lectura i supervisió	60 h	Cap de projecte	18,23	1093,8
<b>Total</b>	<b>440 h</b>			<b>5873,26 €</b>

*Taula 3: costos estimats per activitat*

### 3.5.2 Costos indirectes

S'ha estudiat l'ús dels recursos materials i contractacions de serveis necessaris per portar endavant aquest projecte. Tot seguit es detallen:

- **Amortització del maquinari:** el maquinari emprat en aquest projecte consta principalment d'un equip *Dell Latitude*<sup>12</sup> amb un cost de 800 €. S'espera que la vida útil d'aquest equip sigui de 4 anys i que el 90% de l'ús que en fem serà destinat a aquest projecte.
- **Amortització del programari:** s'utilitzaran eines de desenvolupament lliures i sense cost, s'ha de tenir en compte la infraestructura necessària per mantenir un entorn de desenvolupament en preproducció. Aquesta infraestructura de preproducció constarà d'un servei de màquines virtuals que ens permetrà allotjar el projecte i fer-lo accessible a tot l'equip de desenvolupament, test i client. Aquesta infraestructura virtual funciona amb software lliure i tampoc té cap cost directe associat. S'ha estimat que la infraestructura virtual necessària té un cost de 17,75 €/mensuals que només seran necessaris durant el període de desenvolupament; posteriorment s'utilitzarà la infraestructura de l'Escola de Camins per allotjar el sistema.
- **Connexió de dades:** s'utilitzarà una connexió a Internet de fibra òptica de 300Mbps de *Telefònica* amb un cost de 48 €/mes<sup>13</sup>, del qual el 50% de l'amplada de banda s'utilitzarà al projecte. A més, també es disposarà d'una connexió 3G amb un cost aproximat de 1GB/6€/mes per a realitzar proves amb dispositius *Arduino* remots.
- **Transport:** s'utilitzarà un abonament mensual de 50 viatges, T-50/30 amb un cost de 42,50 €.
- **Impressió a paper:** es consumirà paper majoritàriament pel lliurament final de la Memòria a cadascun dels membres del tribunal, al director i una personal. Estimem una extensió de 200 pàgines aproximadament, amb un cost de 0,05 €/pàgina.

---

<sup>12</sup>[http://www.amazon.es/Dell-Inspiron-5558-Ordenador-port%C3%A1til/dp/B00ZPSK912/ref=sr\\_1\\_15?ie=UTF8&qid=1444154920&sr=8-15&keywords=dell](http://www.amazon.es/Dell-Inspiron-5558-Ordenador-port%C3%A1til/dp/B00ZPSK912/ref=sr_1_15?ie=UTF8&qid=1444154920&sr=8-15&keywords=dell)

<sup>13</sup><http://www.movistar.es/particulares/internet/adsl-fibra-optica/fibra-optica-100mb/?pid=a-fibra-f>

Producte	Unitats	Preu unitari	Dedicació	Cost estimat
Amortització maquinari	4 mesos	800 €/4 anys	90%	59,94 €
Amortització programari	4 mesos	17,75 €/mes	100%	71 €
Connexió dades	4 mesos	54 €/mes	50%	108 €
Transport	4 mesos	42,5 €/mes	100%	170 €
Impressió a paper	1000 pàgines	0,05 €/pàgina	100%	50 €
<b>Total</b>				<b>458,94 €</b>

*Taula 4: costos materials*

### 3.5.3 Contingència

Es reserva una part del pressupost per la partida de contingència, concretament un 20% de la suma de costos directes i un 50% als costos indirectes.

Costos	Percentatge	Preu de mercat	Cost de mercat
Costos directes	20%	5873,26 €	1174,66 €
Costos indirectes	50%	458,94 €	229,47 €
<b>Total</b>			<b>1404,13 €</b>

*Taula 5: contingència de costos directes i indirectes*

### 3.5.4 Imprevistos

- **Retards:** al pla d'acció de la planificació temporal s'indica que es destina un marge de 20 hores extra durant el projecte abans del lliurament final. S'estima que la probabilitat d'entrar en un període de retard és del 40% i la comptabilitzem com hores de programador (9,38 €/hora).
- **Avaria maquinari:** s'estima que la probabilitat d'avaria de l'equip informàtic és d'un 10%.

Imprevist	Probabilitat	Unitats	Preu de mercat	Cost de mercat
Retards	40 %	20 h	9,38 €/h	75,04 €
Avaria maquinari	10 %	1 vegada	800 €	80 €
<b>Total</b>				<b>155,04 €</b>

*Taula 6: costos dels imprevistos*

### 3.5.5 Pressupost

A continuació es mostra l'estimació del pressupost final del projecte.

Concepte	Cost de mercat
Costos directes	5873,26 €
Costos indirectes	458,94 €
Contingència	1404,13 €
Imprevistos	155,04 €
<b>Total</b>	<b>7891,37 €</b>

*Taula 7: pressupost*

### 3.5.6 Control de gestió

Els costos calculats pels diferents perfils necessaris per a aquest projecte s'han basat en articles de referència sobre els tipus de rols i remuneracions d'acord a la seva experiència a les empreses l'any 2015. No obstant això, en aquest projecte tots els rols seran exercits per mi amb una mateixa remuneració, però s'ha volgut identificar els rols concretament per simular un cas real d'una hipotètica empresa.

Es mantindrà un registre d'hores realitzades per tasca per comprovar que realment se segueixen les fites del diagrama de *Gantt*. A cada reunió d'*Sprint Review* es tindrà en compte aquest registre d'hores per identificar possibles desviacions i rectificar-les per ajustar-se a la planificació temporal.

Al final del lliurament es realitzarà un resum real d'hores destinades i costos per comparar-los amb els de la planificació per obtenir els percentatges de desviació i corregir-los per a futurs projectes.

## 4. Anàlisi de requisits

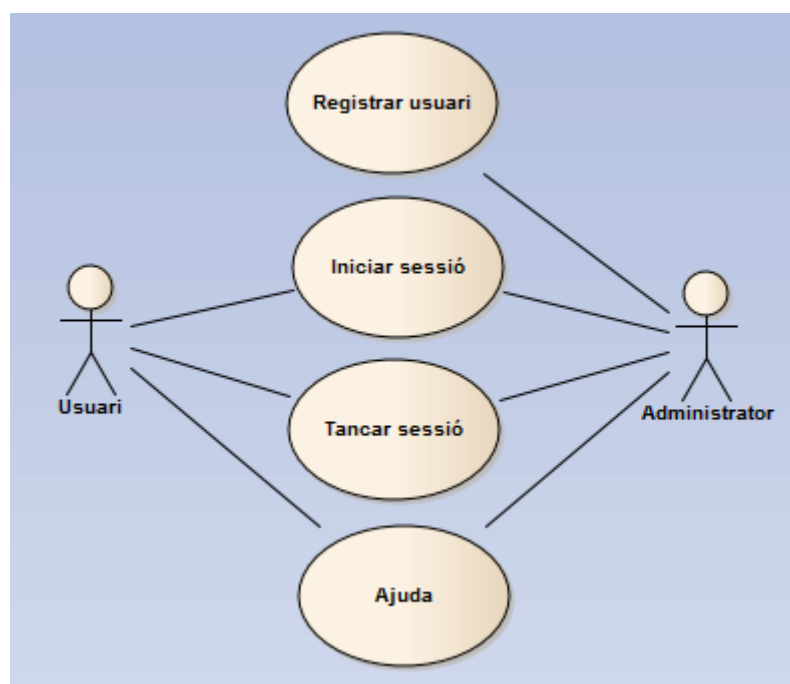
A aquest apartat es definiran els requisits funcionals i no funcionals del sistema que s'han obtingut després d'intenses reunions amb els responsables dels laboratoris i futurs usuaris del sistema. Com ja hem mencionat, existeixen tres rols d'usuari al sistema: Administradors, amb permisos sobre tots els objectes del sistema i la gestió en exclusiva dels usuaris; Responsables de Laboratoris (*Managers*) i Operadors encarregats de publicar dades d'experiments mitjançant dispositius.

### 4.1 Requisits funcionals

Els requisits funcionals són aquells que defineixen el comportament o accions d'un sistema. En aquest apartat definirem els requisits funcionals a implementar especificant els actors als que afecten i com els afecten.

#### 4.1.1 Casos d'ús generals

##### 4.1.1.1 Diagrama



*Il·lustració 18: casos d'ús a la pàgina de benvinguda*

##### 4.1.1.2 CU: Registrar usuari

El primer pas per poder accedir al sistema un cop posat aquest en marxa, és la sincronització dels usuaris del directori de la UPC sol·licitats pels responsables dels laboratoris i assignatures implicades. El personal docent es posarà en contacte amb l'administrador del sistema i li proporcionarà un llistat amb els identificadors dels usuaris de la UPC implicats.

L'Administrador del sistema introduirà aquesta llista a un programari propi de *CaminsTech* que manté sincronitzats les credencials entre l'LDAP de l'Escola de Camins i el sistema web *CaminsSmartLab*.

**Actors:** Administrador.

**Precondició:** -

**Descripció:** Els usuaris del sistema no poden registrar-se per sí mateixos. Un responsable de laboratori, habitualment un docent de l'escola, sol·licitarà a l'Administrador del sistema (responsable de *CaminsTech*) que doni d'alta un seguit d'usuaris de la UPC i assignar-los a un Laboratori. L'Administrador podrà donar d'alta un usuari mitjançant l'eina CLI proporcionant les següents dades:

- Nom d'usuari de la UPC.
- Contrasenya.
- Adreça de correu electrònic.
- Nom complet.
- Rol [Usuari o Administrador]

```
vagrant@smartlab:/vagrant$ sudo php app/console user:create albert.terrone pass@word1 albert.terrone@upc.edu "Albert Terrones" --admin
Created user albert.terrone
vagrant@smartlab:/vagrant$
```

*Il·lustració 19: creació d'un usuari amb rol Administrador per CLI*

**Casos d'error:** els paràmetres anteriors no han estat proporcionats a la línia de comandes o l'usuari ja està registrat.

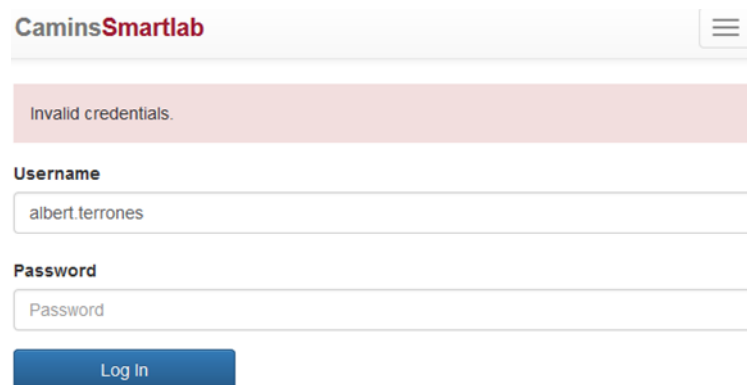
#### 4.1.1.3 CU: Iniciar sessió

**Actors:** Administrador, Usuari.

**Precondició:** l'usuari ha estat registrat prèviament.

**Descripció:** l'usuari podrà iniciar sessió al sistema accedint a l'opció «Log In» i emplenant el formulari amb les seves credencials. Un cop autenticat i autoritzat pel sistema l'usuari serà redirigit a la pàgina principal de Laboratoris modificant la visibilitat de les opcions segons el seu rol.

**Casos d'error:** el nom d'usuari o contrasenya proporcionats no són vàlids.



The screenshot shows the CaminsSmartlab login interface. At the top, the logo 'CaminsSmartlab' is on the left and a menu icon is on the right. Below the logo, a red error message box displays 'Invalid credentials.'. Underneath, there are two input fields: 'Username' containing 'albert.terrone' and 'Password' containing 'Password'. A blue 'Log In' button is positioned below the password field.

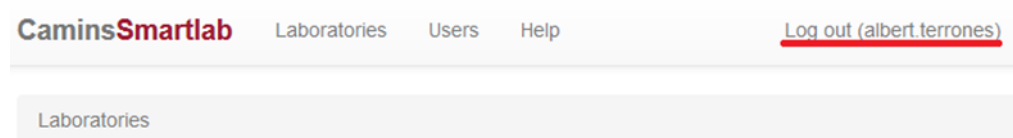
*Il·lustració 20: formulari sol·licitant les credencials amb error d'autenticació*

#### 4.1.1.4 CU: Tancar sessió

**Actors:** Administrador, Usuari.

**Precondició:** l'usuari ha iniciat sessió al sistema.

**Descripció:** l'usuari podrà finalitzar la seva sessió prement l'opció «Log Out», tornant així a la pàgina de benvinguda del CaminsSmartLab.



The screenshot shows the CaminsSmartlab user interface after login. The top navigation bar includes the logo 'CaminsSmartlab', links for 'Laboratories', 'Users', and 'Help', and a red 'Log out (albert.terrone)' link. Below the navigation bar, a 'Laboratories' section is visible.

*Il·lustració 21: «log out» permet tancar sessió*

**Casos d'error:** -



#### 4.1.1.5 CU: Ajuda

**Actors:** Administrador, Usuari.

**Precondició:** -

**Descripció:** l'usuari podrà consultar l'ajuda a qualsevol de les pàgines que es trobi dins del sistema.

**Casos d'error:** -

#### 4.1.2 Casos d'ús de gestió de Laboratori

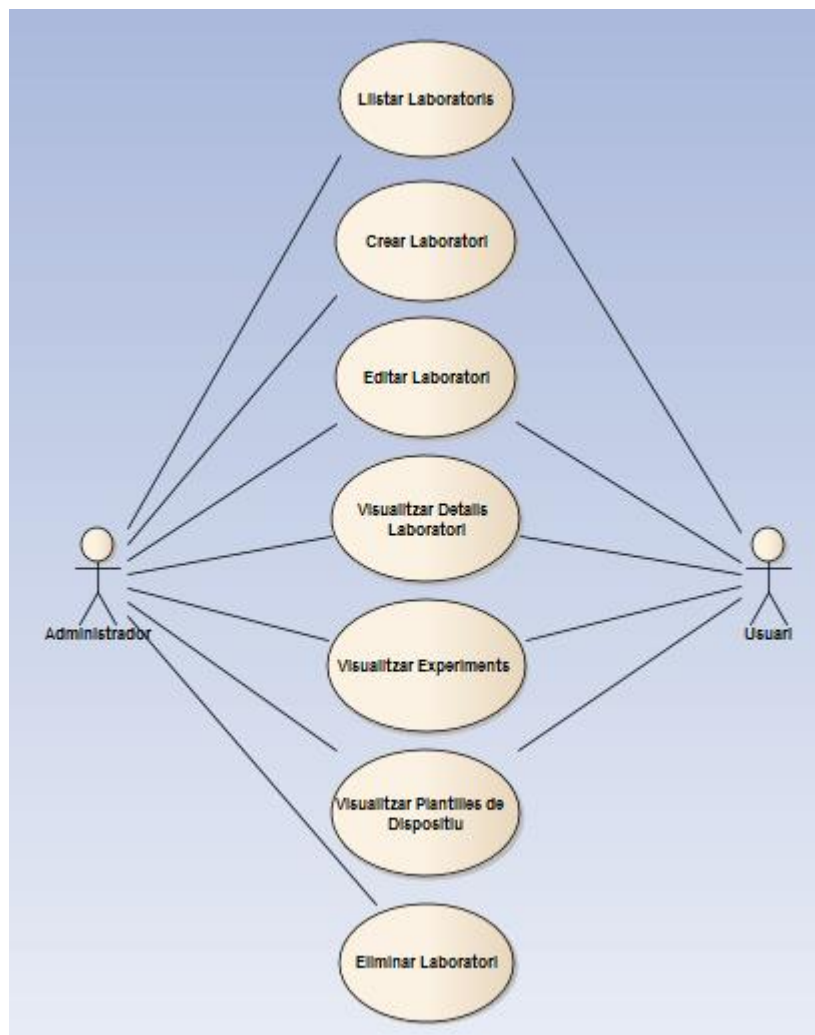
Els Laboratoris són objectes purament informatius amb dades d'interès general, com adreça, especialitat, descripció i persona de contacte d'aquest. Els Laboratoris contenen Experiments, Plantilles de Dispositiu i Dispositius.

Per cada Laboratori es disposarà de:

- Acrònim: codi en l'àmbit de la UPC per referenciar el laboratori. (Ex. LE – Laboratori d'Estructures).
- Nom complet del laboratori.
- Plantilles de dispositiu del laboratori.
- Experiments del laboratori.
- Persona responsable: persona amb permisos per gestionar aquell laboratori.
- Operadors del laboratori: persona amb permisos per publicar Lectures de Dispositiu als experiments del laboratori.

Les operacions sobre aquest tipus d'objectes dependran segons els rols dels usuaris i la seva relació amb el mateix laboratori.

#### 4.1.2.1 Diagrama



*Il·lustració 22: casos d'ús a la pàgina de gestió de laboratoris*

#### 4.1.2.2 CU: Llistar Laboratoris

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** l'Administrador o Usuari pot consultar un llistat de tots els laboratoris del sistema, així com realitzar operacions permeses segons el seu rol sobre aquests laboratoris.

**Casos d'error:** -

#### 4.1.2.3 CU: Crear Laboratori

**Actors:** Administrador.

**Precondició:** l'Administrador ha iniciat sessió al sistema.

**Descripció:** l'Administrador podrà crear laboratoris prement l'opció «*New Laboratory*» i assignar-lo a un usuari registrat convertint-lo a aquest en Responsable del Laboratori (*Laboratory Manager*). També podrà assignar usuaris com a Operadors del Laboratori (*Laboratory Operator*).

**Casos d'error:** les dades introduïdes no són correctes o bé no s'ha seleccionat un Usuari com a Responsable de laboratori.

#### 4.1.2.4 CU: Editar Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és el responsable d'aquest Laboratori.

**Descripció:** l'Administrador o Usuari podrà editar la informació de caràcter públic dels laboratoris o bé assignar-lo a un altre Usuari registrat convertint-lo a aquest en Responsable del Laboratori seleccionant l'opció «*Update*». També podrà assignar usuaris com a Operadors del Laboratori.

**CaminsSmartlab**   Laboratories   Users   Help   [Log out \(albert.terrones\)](#)

Laboratories / LE / Update

**Laboratory fullname**

**Laboratory acronym**

**Specialization**

**Physical Address**

**Manager**

**Operators**

**Short description**

*Il·lustració 23: edició d'informació pública, Responsable i Operadors del Laboratori*

**Casos d'error:** les dades introduïdes no són correctes o bé no s'ha seleccionat un Usuari com a Responsable de laboratori.

#### 4.1.2.5 CU: Visualitzar Detalls de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** l'Administrador o Usuari pot consultar la informació de caràcter públic dels laboratoris, tals com Codi, Nom complet, Nombre d'experiments, Persona responsable i Descripció tal com es mostra al cas d'ús "Editar Laboratori" però en mode lectura.

**Casos d'error:** -

#### 4.1.2.6 CU: Visualitzar Plantilles de Dispositiu de Laboratori

**Actors:** Administrador i Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** l'Administrador o Usuari pot consultar les Plantilles de Dispositiu d'aquell laboratori.

Code	Template Name	Sensors	Devices	Actions
Tmp1	CimentPortland	1	3	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Device Template](#)

Il·lustració 24: Llista de Plantilles de dispositiu (vista Responsable de Laboratori)

**Casos d'error:** -

4.1.2.7 CU: Visualitzar Experiments de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** l'Administrador o Usuari pot consultar els Experiments que es duen a terme dins d'un Laboratori.

Code	Full Name	Last Data received	Open/Closed	Actions
ETCIP30	Experiment sobre la temperatura del Ciment Portland CP30	16-02-2016 17:26:40	yes	<a href="#">Devices (1)</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New experiment](#)

Il·lustració 25: experiments del Laboratori L.E.

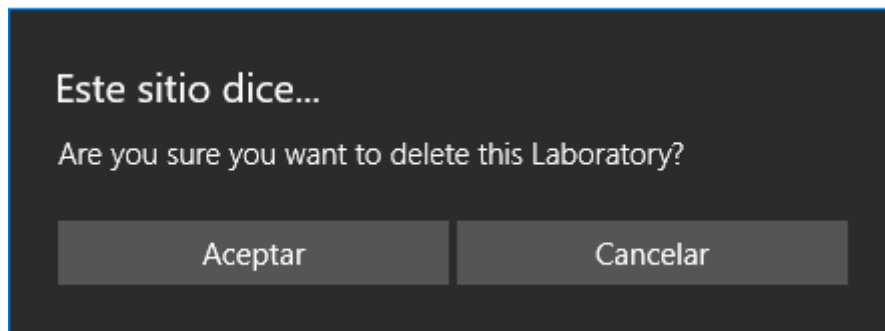
**Casos d'error:** -

#### 4.1.2.8 CU: Eliminar Laboratori

**Actors:** Administrador.

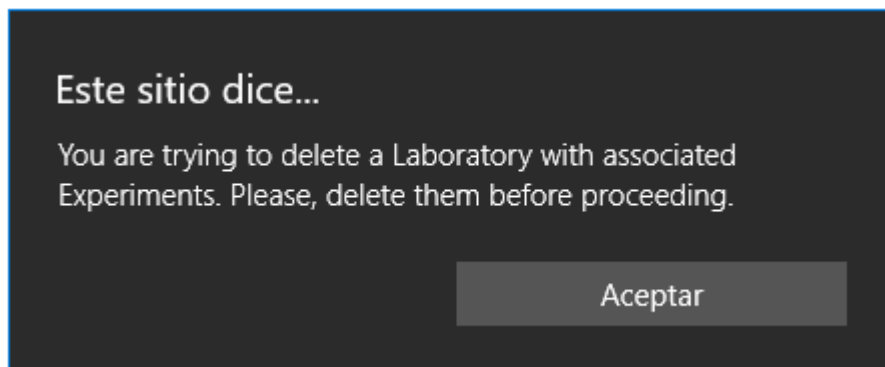
**Precondició:** l'Administrador ha iniciat sessió al sistema. El laboratori no conté experiments.

**Descripció:** l'Administrador podrà eliminar el laboratori sempre que aquest no contingui experiments.



*Il·lustració 26: confirmació d'eliminació d'un Laboratori*

**Casos d'error:** el laboratori conté experiments.



*Il·lustració 27: notificació d'error a l'intentar eliminar un laboratori que conté experiments*

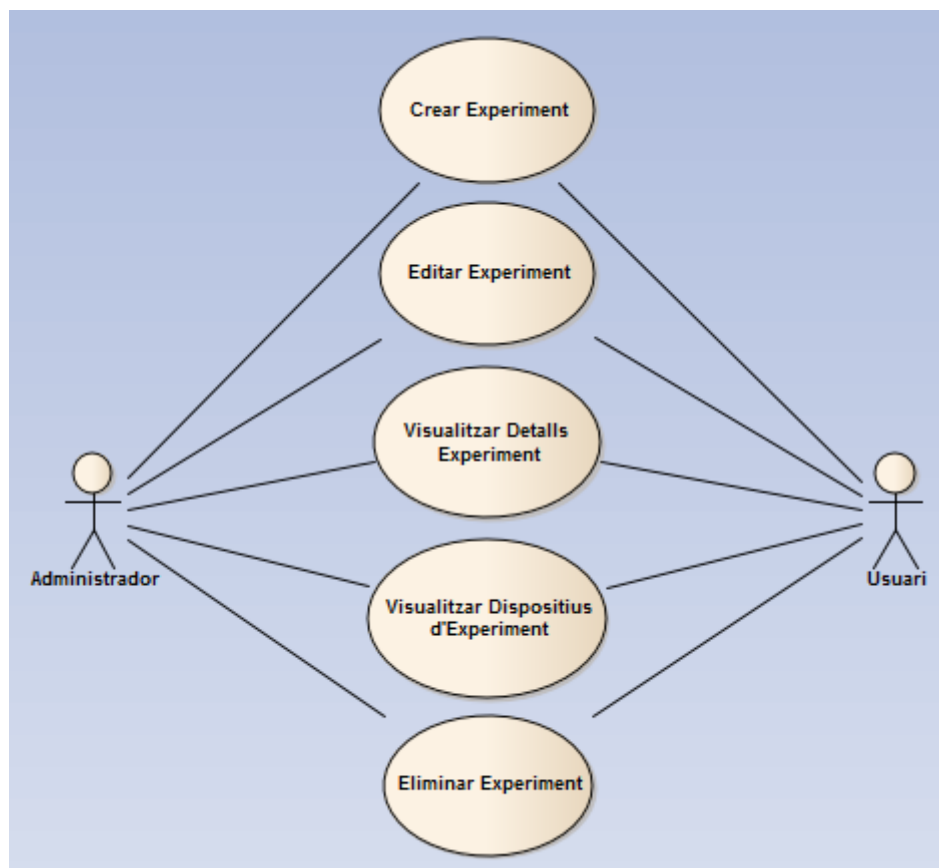
#### 4.1.3 Casos d'ús de gestió d'Experiments de Laboratori

Quan un usuari, amb independència del seu rol, accedeixi a l'opció «*Experiments*» d'un laboratori concret [escenari 4.1.2.7] se li mostrarà la gestió d'experiments d'aquell Laboratori.

El Responsable d'un Laboratori (*Manager*) podrà crear Experiments. Un Experiment és una objecte vinculat a un Laboratori i que conté Dispositius (*Devices*) i una relació amb usuaris Operadors (*Operators*). Aquests Operadors tindran capacitat de gestionar aquests dispositius pels laboratoris als quals estiguin vinculats.

Un Experiment pot tancar-se (*Open/Closed*) en qualsevol moment per part del responsable del laboratori, fet que provoca l'aturada automàtica de captació de dades provinents dels dispositius associats a aquell experiment.

##### 4.1.3.1 Diagrama



*Il·lustració 28: Llistat d'Experiments del Laboratori L.E (vista Manager)*

#### 4.1.3.2 CU: Crear Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

**Descripció:** l'Administrador, però més comunament l'usuari Responsable del Laboratori (*Manager*) podrà crear un Experiment tot assignant-li un nom complet, un codi per referenciar-lo i una breu descripció. L'usuari podrà activar o desactivar l'entrada de dades de sensors d'aquest experiment.

The screenshot shows the 'Create Experiment' form in the CaminsSmartlab application. At the top, there is a navigation bar with the logo 'CaminsSmartlab' and links for 'Laboratories', 'Users', and 'Help'. On the right side of the bar, there is a 'Log out (albert.terrone)' link. Below the navigation bar, a breadcrumb trail reads 'Laboratories / LE / Create Experiment'. The form itself contains several input fields: 'Full Name' (a text box), 'Code' (a text box), and 'Description' (a larger text box). Below these fields is a checkbox labeled 'Accept new Readings', which is currently checked. At the bottom left of the form is a 'Create' button.

*Il·lustració 29: formulari de creació d'un experiment al Laboratori L.E.*

**Casos d'error:** les dades proporcionades no són vàlides o hi ha camps buits.



#### 4.1.3.3 CU: Editar Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

**Descripció:** l'Administrador, però més comunament l'usuari Responsable del Laboratori (*Manager*) podrà editar un Experiment actualitzant el nom complet, el codi per referenciar-lo i la descripció. L'usuari podrà activar o desactivar l'entrada de dades de sensors d'aquest experiment.

The screenshot shows the 'CaminsSmartlab' web application. The top navigation bar includes 'Laboratories', 'Users', 'Help', and a 'Log out (albert.terrone)' link. A breadcrumb trail indicates the path: 'Laboratories / LE / Experiments / ETCIP30 / Update'. The form contains three main sections: 'Full Name' with a text input field containing 'Experiment sobre la temperatura del Ciment Portland CP30'; 'Code' with a text input field containing 'ETCIP30'; and 'Description' with a text area containing 'Experiment sobre la temperatura del Ciment Portland CP30 en situacions de baixes temperatures extremes.'. Below these is a checkbox labeled 'Accept new Readings' which is checked. At the bottom left is an 'Update' button.

*Il·lustració 30: edició de l'Experiment "ETCIP30"*

**Casos d'error:** les dades proporcionades no són vàlides o hi ha camps buits.

#### 4.1.3.4 CU: Visualitzar Detalls d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** permet a l'usuari visualitzar detalls públics de l'experiment, així com la seva data de creació i darrera actualització.

The screenshot shows the 'CaminsSmartlab' web application. At the top, there is a navigation bar with 'Laboratories', 'Users', and 'Help' links, and a 'Log out (albert.terrone)' button. Below this is a breadcrumb trail: 'Laboratories / LE / Experiments / ETCIP30 / Details'. The main content area displays the following details for experiment ETCIP30:

- Code:** ETCIP30
- Open to new readings:** yes
- Create time:** 12-02-2016 21:18:47
- Update time:** 29-02-2016 12:35:19
- Description:** Experiment sobre la temperatura del Ciment Portland CP30 en situacions de baixes temperatures extremes.

*Il·lustració 31: detalls de l'Experiment "ETCIP30" del Laboratori L.E.*

**Casos d'error:** -

#### 4.1.3.5 CU: Visualitzar Dispositius d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** permet a l'usuari visualitzar el llistat de Dispositius actius que proporcionen dades de sensors a l'experiment.

The screenshot shows the 'CaminsSmartlab' web application. At the top, there is a navigation bar with 'Laboratories', 'Users', and 'Help' links, and a 'Log out (albert.terrone)' button. Below this is a breadcrumb trail: 'Laboratories / LE / Experiments / ETCIP30 / Devices'. The main content area displays a table of devices:

Code	Fullname	Last reading	Template	Actions
3	Arduino de l'Albert	16-02-2016 17:26:40	CimentPortland	<a href="#">Download</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

Below the table, there is a green button labeled 'New Device'.

*Il·lustració 32: llistat de dispositius de l'experiment "ETCIP30" al Laboratori L.E.*

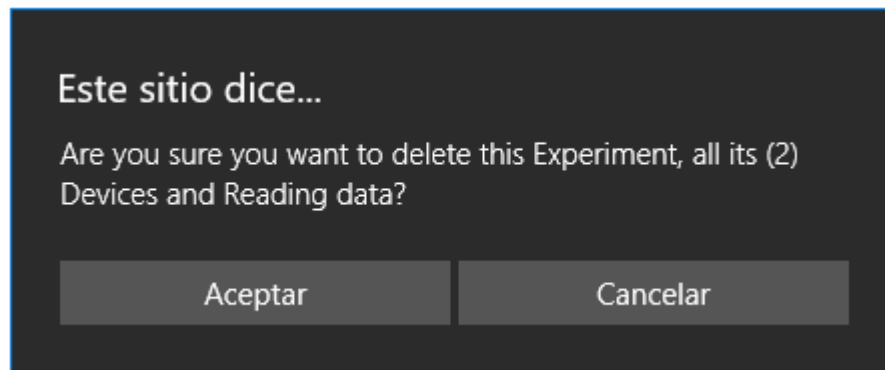
**Casos d'error:** -

#### 4.1.3.6 CU: Eliminar Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

**Descripció:** permet eliminar un Experiment, totes les plantilles de dispositiu, dispositius i dades de sensors en cascada.

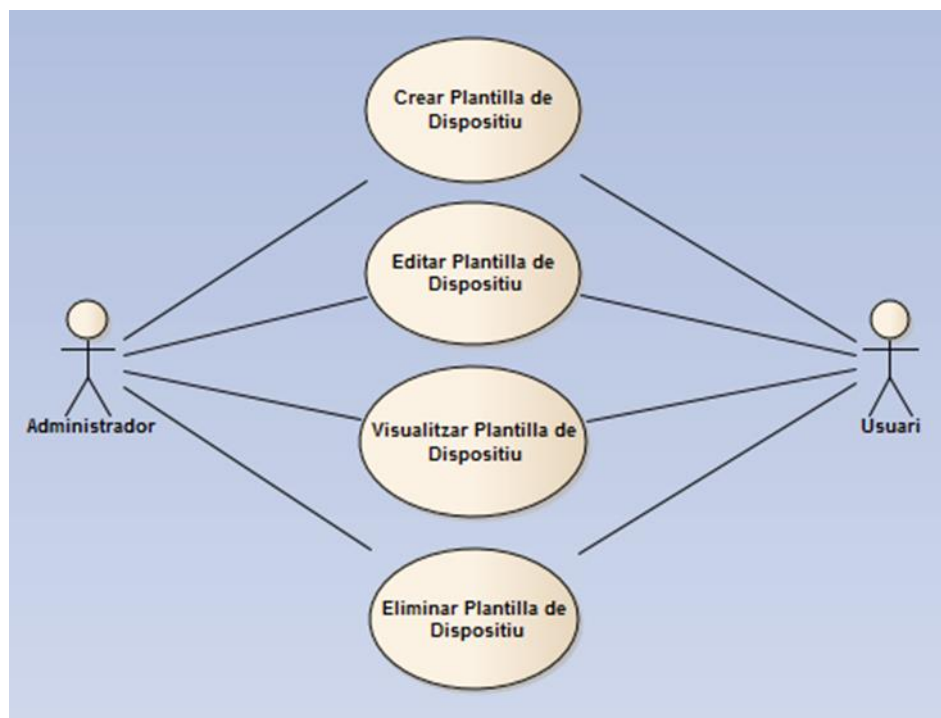


*Il·lustració 33: eliminació d'un experiment i totes les seves dades associades*

**Casos d'error:** -

#### 4.1.4 Casos d'ús de gestió de Plantilles de Dispositiu de Laboratori

##### 4.1.4.1 Diagrama



*Il·lustració 34: casos d'ús a la pàgina de gestió de Plantilles de Dispositiu*

#### 4.1.4.2 CU: Crear Plantilla de Dispositiu de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

**Descripció:** permet crear una Plantilla de Dispositiu a un Laboratori, permetent la reutilització de dispositius en posteriors experiments d'aquell laboratori.

The screenshot shows the 'CaminsSmartlab' web application interface. At the top, there is a navigation bar with links for 'Laboratories', 'Users', and 'Help', and a 'Log out (albert.terrones)' button. Below the navigation bar, a breadcrumb trail indicates the current path: 'Laboratories / LE / Create Device Template'. The main form is titled 'Code' and contains several input fields: 'Template Name' (containing 'Tmp1'), 'Sensor Name 1' (containing 'CimentPortland'), 'Sensor Name 2' (containing 'temperature'), 'Sensor Name 3' (containing 'humidity'), and 'Sensor Name 4' (containing 'date\_time'). There is a '+' button to add more sensors. At the bottom of the form is a 'Create Template' button.

*Il·lustració 35: creació d'una Plantilla de Dispositiu amb N sensors*

**Casos d'error:** el codi de plantilla proporcionat ja existeix o les dades introduïdes no són vàlides.

#### 4.1.4.3 CU: Editar Plantilla de Dispositiu de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

**Descripció:** permet editar una Plantilla de Dispositiu d'un Laboratori canviant informació d'interès, com el codi i nom de la plantilla, així com el tipus de sensors d'aquella plantilla. El contingut de la vista és pràcticament idèntic al cas d'ús de "Creació" amb canvis visuals de «Create» per «Update».

**Casos d'error:** el codi de plantilla modificat ja existeix o les dades introduïdes no són vàlides.

#### 4.1.4.4 CU: Visualitzar Plantilla de Dispositiu de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** permet consultar les dades d'una Plantilla de Dispositiu per conèixer-ne el seu nom i els sensors que defineixen aquesta plantilla, útil per a crear un nou dispositiu. El contingut de la vista és pràcticament idèntic al cas d'ús de "Creació" amb canvis visuals i camps de només lectura.

**Casos d'error:** -

#### 4.1.4.5 CU: Eliminar Plantilla de Dispositiu de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és responsable d'aquest Laboratori.

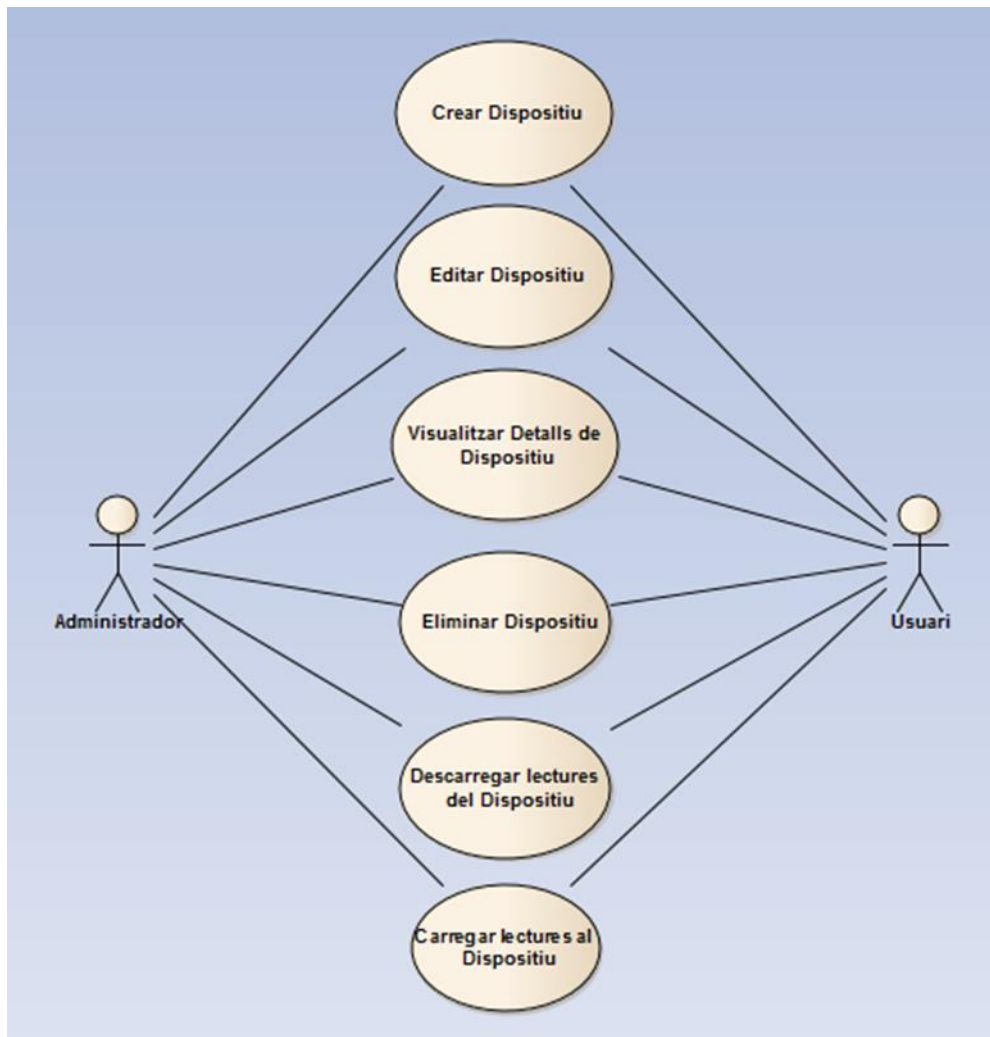
**Descripció:** permet eliminar una Plantilla de Dispositiu sempre que no hi hagi Dispositius utilitzant-la.

**Casos d'error:** la plantilla de dispositiu està essent utilitzada per un Dispositiu actualment.

#### 4.1.5 Casos d'ús de gestió de Dispositius d'Experiment de Laboratori

Els Dispositius o recursos de dades (*Devices*) són objectes que representen pròpiament els dispositius físics que s'utilitzen en un Laboratori per a un Experiment. Cada un d'aquests dispositius està vinculat a un *DeviceTemplate*; l'Operador haurà de programar el dispositiu físic amb els sensors seguint les especificacions de la Plantilla de Dispositiu vinculada, així com també a l'hora de publicar dades d'un experiment mitjançant un fitxer CSV.

##### 4.1.5.1 Diagrama



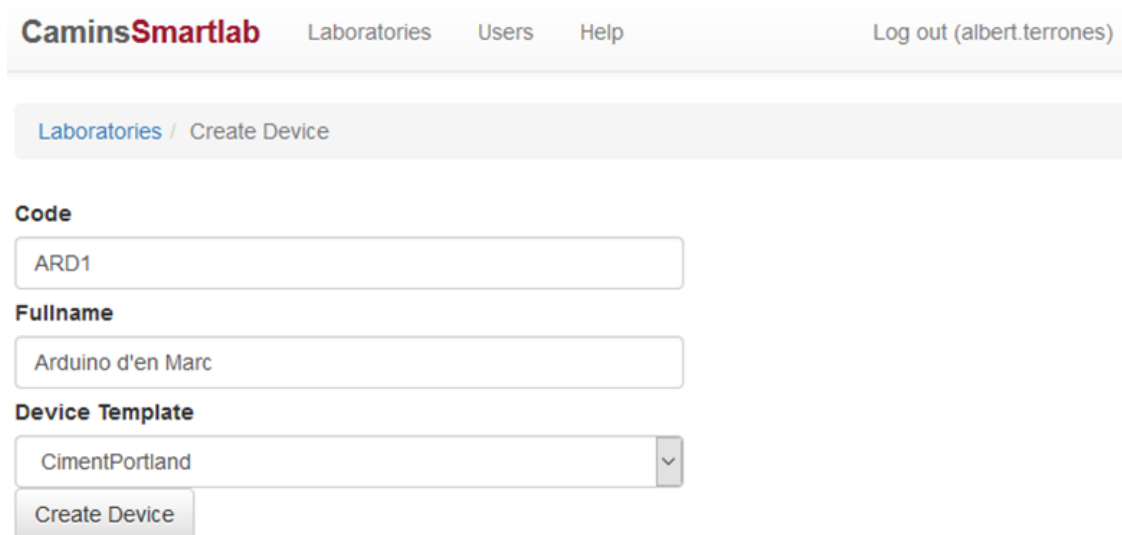
*Il·lustració 36: casos d'ús a la pàgina de gestió de Dispositius d'Experiments de Laboratori*

#### 4.1.5.2 CU: Crear Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** permet a un usuari associat al Laboratori donar d'alta un Dispositiu físic basant-se en una Plantilla de Dispositiu del propi Laboratori.



The screenshot shows the 'CaminsSmartlab' web interface. At the top, there is a navigation bar with links for 'Laboratories', 'Users', and 'Help', and a 'Log out (albert.terrone)' button. Below the navigation bar, a breadcrumb trail indicates 'Laboratories / Create Device'. The main form is titled 'Code' and contains three input fields: 'Code' (with the value 'ARD1'), 'Fullname' (with the value 'Arduino d'en Marc'), and 'Device Template' (a dropdown menu with 'CimentPortland' selected). A 'Create Device' button is located at the bottom of the form.

*Il·lustració 37: creació d'un Dispositiu basat en la Plantilla de Dispositiu "CimentPortland"*

**Casos d'error:** el codi de dispositiu proporcionat ja existeix, les dades introduïdes no són vàlides o no s'ha seleccionat cap Plantilla de Dispositiu.

#### 4.1.5.3 CU: Editar Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema.

**Descripció:** permet editar el codi, nom del dispositiu i amb quina Plantilla de Dispositiu està basat el dispositiu.

**Casos d'error:** el codi de dispositiu proporcionat ja existeix, les dades introduïdes no són vàlides o no s'ha seleccionat cap Plantilla de Dispositiu. No es pot canviar la Plantilla de Dispositiu si ja hi ha lectures de sensors enregistrades.

#### 4.1.5.4 CU: Visualitzar Detalls de Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és col·laborador del Laboratori.

**Descripció:** permet a l'usuari consultar les dades del dispositiu, com el codi físic i el nom però també les dades de connexió per al dispositiu, com l'adreça HTTP per a realitzar peticions a la REST API i el *token* per autenticar el dispositiu. No menys important, també es mostren les lectures de sensors obtingudes pel dispositiu físic en forma de taula.

**CaminsSmartlab** Laboratories Users Help Log out (albert.terrones)

### Arduino de l'Albert

**Code**  
3

**Fullname**  
Arduino de l'Albert

**Device Template**  
CimentPortland

**Create time**  
12-02-2016 21:19:38

**Last Data input**  
2016-02-21 18:30

**REST API: PUT METHOD**  
https://smartlab.camintech.upc.edu/api/readings/3

**Token**  
e56be4c6a7be8a

**Device Data**

Show 10 entries Search:

temperature	humidity	date_time
4	5	2016-02-21 18:34
4	5	2016-02-21 18:33
4	5	2016-02-21 18:32
4	5	2016-02-21 18:31

Showing 1 to 4 of 4 entries Previous 1 Next

Export to CSV

*Il·lustració 38: informació completa d'un dispositiu amb lectures de sensors*



#### 4.1.5.5 CU: Eliminar Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és col·laborador del Laboratori.

**Descripció:** permet eliminar una Dispositiu i les seves dades de sensors.

**Casos d'error:** -

#### 4.1.5.6 CU: Descarregar lectures de Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és col·laborador del Laboratori.

**Descripció:** permet descarregar en un fitxer CSV el conjunt de dades de sensors d'un dispositiu.

**Casos d'error:** -

#### 4.1.5.7 CU: Carregar lectures de Dispositiu d'Experiment de Laboratori

**Actors:** Administrador, Usuari.

**Precondició:** l'Administrador o Usuari ha iniciat sessió al sistema. L'Usuari és col·laborador del Laboratori.

**Descripció:** permet carregar i afegir dades de sensors mitjançant un fitxer CSV al conjunt de lectures de sensors (*Readings*) d'un dispositiu. Especialment útil en experiments *offline* que es realitzen mitjançant fitxers CSV i no es pot utilitzar la REST API del sistema per falta de connectivitat o rendiment de la xarxa.

**Casos d'error:** el fitxer CSV no compleix les especificacions definides per la plantilla del dispositiu associada.

## 4.2 Requisits no funcionals

Els requisits no funcionals són aquells que permeten el bon funcionament del *software*. Aquests tenen en compte requisits de qualitat com l'eficiència, el rendiment, la seguretat i la fiabilitat del *software*.

A continuació enumerem els requisits no funcionals que el sistema *software* satisfà i la tècnica emprada per assolir-ho:

1. **El sistema ha de ser de fàcil ús per a qualsevol usuari:** el sistema web està principalment dirigit a usuaris amb coneixements tècnics bàsics relatius a les aplicacions web. El sistema disposa d'una interfície molt simple i intuïtiva que els permet entendre l'estructura organitzativa de l'aplicació. Per l'Administrador, la gestió d'usuaris es realitza a través d'una eina CLI realitzada sota la seva petició, fet que també li facilita aquesta tasca a l'hora d'automatitzar el procés mitjançant d'*scripts*.
2. **L'idioma del sistema ha d'estar inicialment en anglès (EN-en):** el sistema web haurà d'estar inicialment en anglès (ISO639-1 EN-en) amb la possibilitat d'afegir nous idiomes en un futur, fet que ja permet *Symfony*. S'ha decidit fer-lo en anglès ja que molts alumnes en pràctiques són d'Erasmus i així es facilita l'operació del sistema a tothom.
3. **El sistema ha de tenir una estètica estàndard:** *CaminsSmartLab* ha de seguir les pautes de disseny imposades a totes les aplicacions de *CaminsTech*. Segueix els mateixos dissenys i recursos gràfics comuns en altres aplicacions, així com el tipus de font, colors i logotips institucionals.
4. **El sistema web ha de ser accessible:** *CaminsSmartLab* compleix amb els estàndards W3C i incorpora mecanismes de disseny responsiu per adaptar-se correctament a qualsevol dispositiu, així com satisfer la configuració de visualització del navegador de l'usuari.
5. **Interfície d'aplicació estàndard:** la interfície (API) dispositiu – servidor web ha de ser estandarditzada i permetre la interoperabilitat independentment del tipus de dispositiu utilitzat (HTTP i RESTful API).
6. **Baix temps de resposta:** totes les peticions realitzades tant per part de l'usuari com dels dispositius s'han de poder resoldre pel sistema en pocs segons (<5seg.). Queda exempt d'aquest requisit el tractament de fitxers CSV que pot requerir més temps en funció del volum de dades a tractar, sense arribar a superar els 5 minuts en aquest últim cas.

7. **Protecció de la informació:** el sistema ha de garantir la protecció de les dades mitjançant autenticació i controls d'accés a aquestes als usuaris autoritzats dins de l'àmbit de l'Escola de Camins.

Les credencials són sincronitzades amb el sistema CAS de l'Escola i en cap moment es disposa d'informació sensible, com les contrasenyes, dels usuaris en text pla.

A més, és requisit indispensable que el sistema web estigui sota el control de *CaminsTech*, i per tant no pot allotjar-se en infraestructures ni serveis externs. El sistema ha de complir amb la LOPD.

8. **El sistema ha de garantir alta disponibilitat:** desenes de dispositius poden estar enviant dades de sensors útils per experiments de l'Escola de Camins a una freqüència màxima de 2 peticions/dispositiu/minut, una baixa disponibilitat pot provocar que els experiments no es puguin realitzar per la pèrdua parcial o completa de la informació proporcionada pels sensors. És per aquest motiu que cal garantir-ne la disponibilitat.

El projecte web serà desplegat en un entorn virtual de la UPC anomenat *Cloud UPC*. Aquest entorn garanteix una alta disponibilitat i seguretat de la mateixa. El fet d'ubicar-se en un sistema virtualitzat permet garantir la disponibilitat del sistema al assignar recursos físics (RAM, CPU, emmagatzematge i ample de banda) dinàmicament segons la demanda.

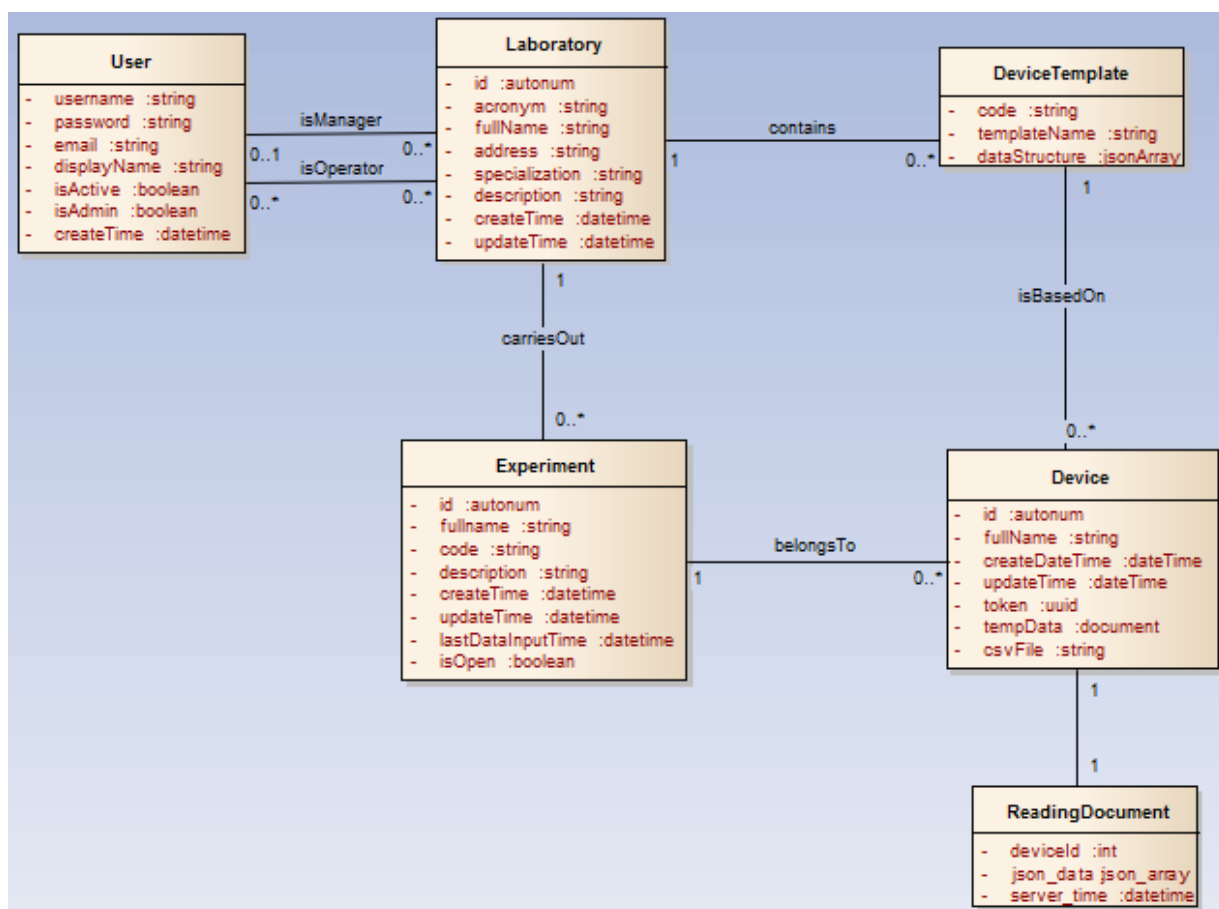
9. **El sistema software ha de tenir suport tècnic a llarg termini:** *Symfony* està activament desenvolupat per *SensioLabs* i es proporcionen dues branques<sup>14</sup> de desenvolupament, la *fast* amb actualitzacions constants però menor estabilitat, o bé la LTS amb un desenvolupament més lent però més estable. Pel nostre projecte utilitzarem sempre la branca de ***Symfony2.8*** que correspon a la LTS amb suport fins l'any 2018.

---

<sup>14</sup> <http://symfony.com/doc/current/contributing/community/releases.html#long-term-support-versions>

## 4.3 Model conceptual de dades

En aquest apartat es descriu el Model de Dades dissenyat mitjançant un diagrama de classes UML i que modela la realitat dels laboratoris de l'Escola de Camins.



Il·lustració 39: model de dades de CaminsSmartLab

### 4.3.1 Descripció textual del model conceptual de dades

Els Laboratoris de l'Escola de Camins (*Laboratory*) són unitats que contenen dades de caràcter informatiu del laboratori com el seu nom complet (*fullname*), acrònim (*acronym*), adreça física (*physical address*), especialitat (*specialization*) i una descripció (*description*). A més hi emmagatzemem la data de creació i l'última modificació (*createTime* i *updateTime*).

Els Laboratoris contenen Usuaris (*User*) que són entitats sincronitzades mitjançant un sistema extern de CaminsTech i fa referència als usuaris de l'àmbit de la UPC que tenen alguna relació amb el sistema. Hi desem informació bàsica d'identificació. Els Usuaris es relacionen amb Laboratori mitjançant una relació de Responsable de Laboratori (rel: *isManager*) i una altra d'Operador de Laboratori (rel: *isOperator*). Un Usuari pot definir-se com a Administrador (*isAdmin*) i estar actiu o inactiu (*isActive*).

Un Laboratori conté Experiments (*Experiment*). Aquests són unitats que descriuen un experiment que es realitza dins d'un laboratori (rel: *carriesOut*) i és definit pel

Responsable del laboratori. L'experiment conté informació bàsica d'informació com el nom, descripció i dates de creació i actualització. Un Experiment pot estar obert o tancat; en cas d'estar tancat no acceptarà noves dades i només estarà disponible en mode de consulta.

Un Laboratori també conté Plantilles de Dispositiu (*Device Template*) (rel: *contains*). Aquestes són unitats definides pel responsable de laboratori i que permeten definir un tipus de dispositiu mitjançant l'especificació dels sensors que conté aquest tipus de dispositiu. Permet validar la informació publicada pels dispositius i generar excepcions si les dades publicades per un dispositiu no corresponen a les definides per la seva plantilla. A més, permeten reaprofitar els dispositius físics per a futurs experiments sense haver de redefinir-los. Una Plantilla de Dispositiu conté codi públic, nom, i els camps dels sensors (*dataStructure*).

Un Experiment conté Dispositius (*Devices*). Els dispositius són únics per un Experiment (rel: *belongsTo*) i estan basats en una Plantilla de Dispositiu (rel: *isBasedOn*). Un dispositiu d'un experiment conté informació pública com el seu codi i nom, data de creació i última modificació, nom del fitxer CSV opcional, i un camp d'emmagatzematge de lectures (*tempData*). Un Dispositiu conté un *token* autogenerat que serveix de clau a l'hora de realitzar peticions a la REST API.

Un Dispositiu (*Device*) conté un *ReadingDocument*. Els *ReadingDocument* són elements que contenen en una estructura de dades JSON, les dades dels sensors publicades pels Dispositius. Cada Dispositiu conté un *ReadingDocument*.

## 5. Disseny final de la interfície

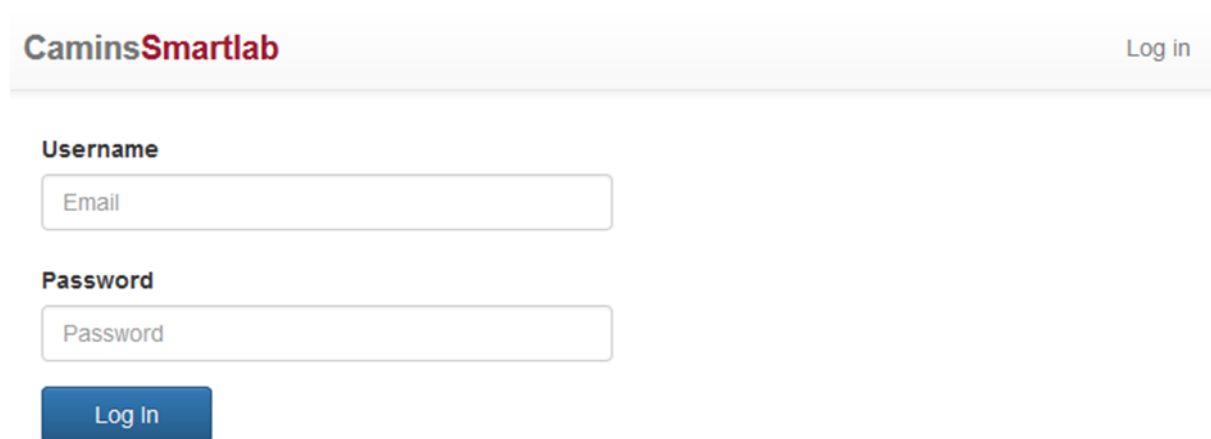
---

A continuació es mostren els dissenys finals del projecte. Aquests s'han creat seguint els principis de disseny establerts per *CaminsTech* per a totes les aplicacions desenvolupades. S'han seguit els logotips, fonts i colors corporatius per donar una imatge de cohesió entre totes les aplicacions desenvolupades per *CaminsTech*.

Les funcionalitats i ubicació dels components s'ha mantingut respecte als dissenys inicials explicats a [3.3 – Abast del Projecte](#).

### 5.1 Pàgina principal (Benvinguda i Inici de sessió)

Aquesta és la pàgina<sup>15</sup> que es mostrarà a tots els usuaris quan accedeixin al sistema tant si pertanyen com si no a l'Escola de Camins. És la part pública de la web.



CaminsSmartlab Log in

**Username**

**Password**

Log In

*Il·lustració 40: formulari d'inici de sessió mitjançant les credencials de la UPC*

---

<sup>15</sup> <https://smartlab.camintech.upc.edu>

## 5.2 Gestió de Laboratoris

A continuació es visualitza la pàgina que es mostra als usuaris autenticats de l'Escola de Camins després d'haver validat les seves credencials.

CaminsSmartlab

LaboratoriesUsersHelp

Log out (albert.terrones)

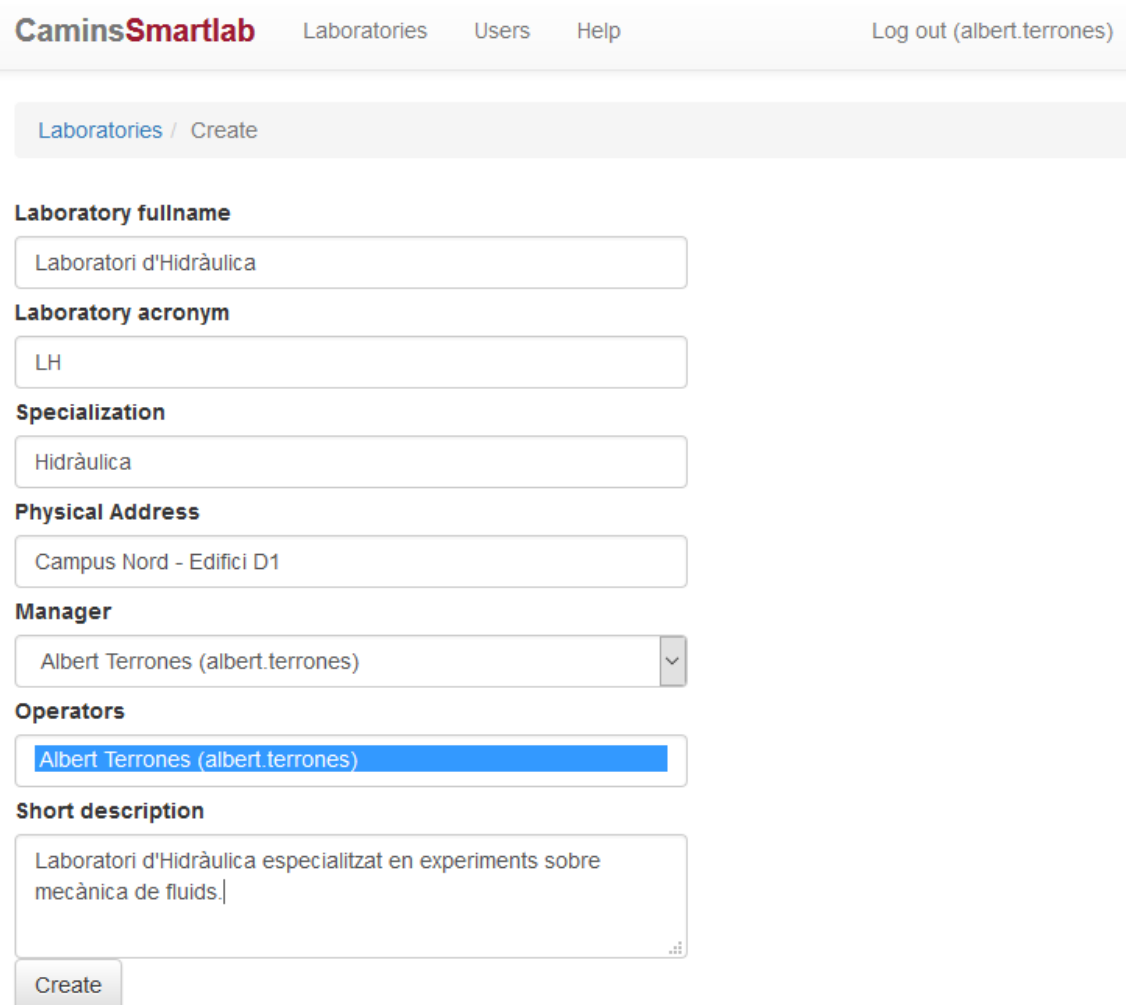
Laboratories

Acronym	Full Name	Experiments	Manager	Actions
LE	Laboratori d'Estructures	1	albert.terrones	<div>Experiments (1)DetailsUpdateDelete</div>
LH	Laboratori d'Hidràulica	0	albert.terrones	<div>Experiments (0)DetailsUpdateDelete</div>

New laboratory

*Il·lustració 41: llistat de Laboratoris*

Aquesta captura mostra com és la creació d'un laboratori. L'edició i la visualització de detalls són pràcticament idèntiques.



CaminsSmartlab   Laboratories   Users   Help   Log out (albert.terrones)

Laboratories / Create

**Laboratory fullname**  
Laboratori d'Hidràulica

**Laboratory acronym**  
LH

**Specialization**  
Hidràulica

**Physical Address**  
Campus Nord - Edifici D1

**Manager**  
Albert Terrones (albert.terrones) ▼

**Operators**  
Albert Terrones (albert.terrones)

**Short description**  
Laboratori d'Hidràulica especialitzat en experiments sobre mecànica de fluids.]

Create

*Il·lustració 42: creació, Edició i Visualització de Laboratori*

### 5.3 Gestió d'Experiments d'un Laboratori

Si es selecciona l'opció «*Experiments*» d'un laboratori concret es mostra a l'usuari la gestió d'Experiments d'aquell Laboratori.

Code	Full Name	Last Data received	Open/Closed	Actions
ETCIP30	Experiment sobre la temperatura del Ciment Portland CP30	16-02-2016 17:26:40	yes	<a href="#">Devices (1)</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New experiment](#)

*Il·lustració 43: gestió d'Experiments d'un Laboratori (L.E.)*

### 5.4 Gestió de Plantilles de Dispositiu d'un Laboratori

Si es selecciona l'opció «*Device Templates*» d'un laboratori concret es mostra a l'usuari la gestió de Plantilles de Dispositiu d'aquell Laboratori.

Code	Template Name	Sensors	Devices	Actions
Tmp1	CimentPortland	1	3	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Device Template](#)

*Il·lustració 44: gestió de Plantilles de Dispositiu d'un Laboratori (L.E.)*

### 5.5 Gestió de Dispositius d'un Laboratori

Si es selecciona l'opció «*Devices*» d'un experiment d'un laboratori es mostra a l'usuari la gestió de Dispositius d'aquell experiment.

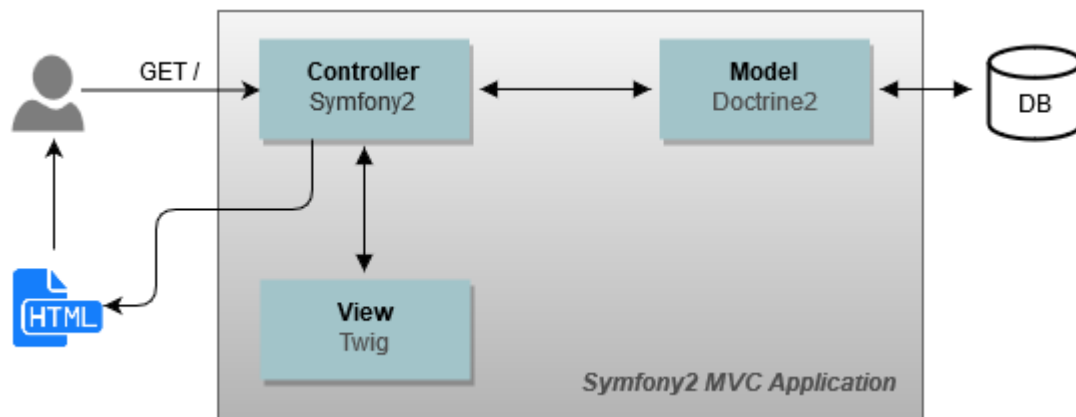
Code	Fullname	Last reading	Template	Actions
3	Arduino de l'Albert	16-02-2016 17:26:40	CimentPortland	<a href="#">Download</a> <a href="#">Details</a> <a href="#">Update</a> <a href="#">Delete</a>

[New Device](#)

*Il·lustració 45: gestió de Dispositius de l'Experiment "ETCIP30" del Laboratori L.E.*



## 6. Arquitectura i implementació de *CaminsSmartLab*



Il·lustració 46: arquitectura MVC d'aplicació tradicional a Symfony2

*Symfony*<sup>16</sup> és un *framework* d'aplicacions web basat en PHP molt potent que cerca sobretot la màxima organització i modularitat del codi en el paradigma Model-Vista-Controlador (MVC). Cerca el RAD (*Rapid Application Development*) mitjançant mòduls o *bundles* que proporcionen funcionalitats habituals i així afavorir la reutilització de codi. Alguns exemples són la gestió d'usuaris i creació de REST APIs. Aquests *bundles* poden ser distribuïts entre la comunitat del software lliure i faciliten així la reutilització de codi entre projectes.

*Symfony* es caracteritza per accelerar la creació i el manteniment d'aplicacions web en PHP (RAD), baix acoblament dels components, alta personalització i ser fàcilment validable.

*Symfony* ha sigut dissenyat per satisfer els següents requisits:

- Facilitat d'instal·lació i configuració en multiplataforma.
- Independència del motor de base de dades escollit.
- Facilitat d'ús però suficientment flexible per satisfer casos complexos.
- Satisfà les bones pràctiques i patrons de disseny en aplicacions web.
- Estabilitat per projectes de llarga durada (LTS).
- Fàcil manteniment i integració amb llibreries de tercers.

Moltes de les característiques habituals dels projectes web són automatitzats amb *Symfony*:

- Mecanisme d'internacionalització per multiidioma.
- La capa de presentació treballa mitjançant plantilles que poden ser desenvolupades per dissenyadors d'HTML sense coneixement del *framework*.

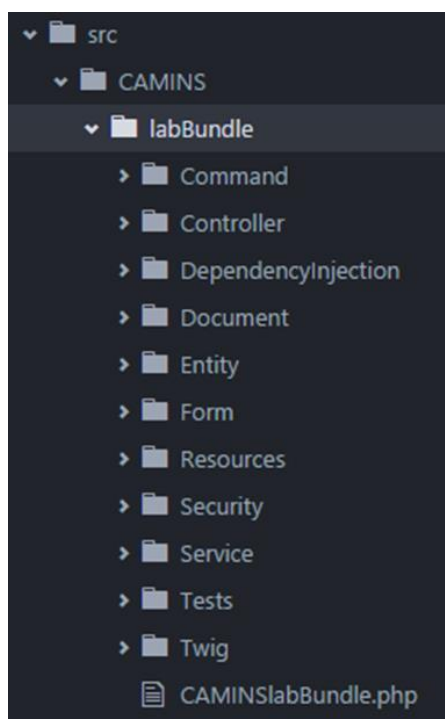
---

<sup>16</sup> <https://symfony.com/what-is-symfony>

- Creació automàtica de formularis per facilitar la validació de les dades i assegurar així la qualitat d'aquestes a la base de dades.
- Funcionalitats de cache per reduir l'ús de l'amplada de banda i la càrrega del servidor.
- Gestió de seguretat per l'autenticació i autorització dels usuaris.
- Enrutament i URL amigables.
- Mecanismes de paginació, ordenació i filtratge.

## 6.1 Bundle

Els *bundles*<sup>17</sup> són el nom que s'utilitza a *Symfony* per descriure un "lloc web" i que es caracteritza en si mateix com un paquet. Tot el codi de la nostra aplicació s'emmagatzema dins de *bundles*. Així mateix, si volguéssim construir un Fòrum dins del nostre lloc web, podríem crear un *bundle* "Fòrum" que fos independent del *bundle* "e-commerce" del nostre sistema web. Aquests *bundles* creen el que podríem dir, tota la nostra aplicació. Els *bundles* contenen una estructura de directoris proposada per *Symfony* (però no obliguen) per desenvolupar la nostra aplicació seguint l'arquitectura MVC i alhora organitzar el codi en directoris lògics. Els *bundles* són fàcilment portables i convertibles en *Vendors* per ser distribuïts i reinstal·lats per qualsevol desenvolupador en forma de paquet a la seva pròpia aplicació.



Il·lustració 47: bundle "labBundle" de l'organització "CAMINS"

<sup>17</sup> <https://knpuniversity.com/blog/AppBundle>

A la nostra aplicació tenim únicament un *bundle* anomenat “*labBundle*” dins de l’organització “*CAMINS*” en el qual hem organitzat tot el codi del nostre sistema web.

Com es veurà en detall posteriorment, aquest *bundle* inclou tots els components essencials per satisfer els patrons de l’arquitectura MVC (entitats, controladors, vistes...) però també altres recursos útils com la definició de normes de seguretat, serveis de bases de dades i tests.

*Symfony* ens proveeix d’un assistent per crear aquests *bundles* de manera semiautomàtica mitjançant la línia de comandes:

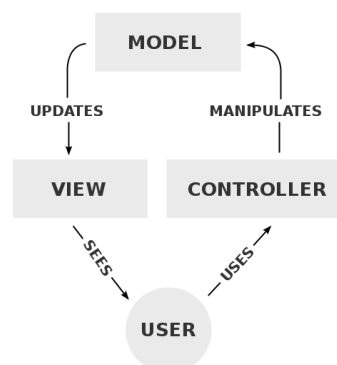
```
php app/console generate:bundle --namespace=CAMINS/labBundle --format=yml
```

*Codi 1: creació d'un bundle per CLI*

## 6.2 Arquitectura MVC

L’arquitectura de tres capes MVC[9] o Model Vista Controlador consisteix en separar la lògica de negoci de la presentació de la informació mitjançant tres capes amb responsabilitats ben diferents:

- **Model:** és la capa on es defineixen les entitats d’informació que estem reflectint del món real. Aquesta capa es troba just per sobre de la base de dades i per tant realitza el mapeig entre aquesta i els objectes del món real amb què treballarem.
- **Vista:** és la capa més propera a l’usuari final, ja que és l’encarregada de mostrar la informació del sistema a l’usuari o agents externs.
- **Controlador:** és la capa encarregada de rebre i processar les peticions rebudes per part de l’usuari o agents externs (una REST API, per exemple). El controlador rep la petició externa i envia comandes al model per actualitzar el seu estat; una vegada modificat l’estat, envia comandes a la capa de la Vista per actualitzar la presentació visual del model.



*Il·lustració 48: interacció entre els components del MVC*

A *Symfony*, aquests components es troben repartits en diferents paquets tal com s'explica a continuació:

- **Resources/Views:** correspon a la capa de *Vista*. En aquesta s'hi ubiquen els elements visuals propis del *bundle* que es presentaran a l'usuari en forma de codi HTML, *JavaScript* i CSS. Aquests elements es comuniquen amb la capa de Controlador mitjançant un component extern de *Symfony* anomenat *Twig* que facilita la transferència de dades entre ambdues capes.
- **Controller:** Correspon, com el seu nom indica, a la mateixa capa de *Controlador*. S'encarrega de rebre les peticions HTTP de l'usuari o procedents de la REST API, gestiona i valida aquestes peticions, i envia ordres als *Entity*, *Document*, *Form* i *Service* per operar amb els models i retornar una resposta d'acord a la petició.
- **Entity/Document/Form/Service:** En aquest conjunt de blocs es reparteix tota la capa de *Model* del MVC. A *Symfony*, una classe del món real és representada per una *Entity*. En el nostre cas, també treballem amb el concepte més tècnic de *Document* que té una representació i manipulació de persistència diferent dels *Entity* com s'explicarà més endavant. Cada *Entity* té associat un *Form* que defineix amb quins atributs voldrem treballar a la capa de la *Vista*, i per últim, al bloc de *Service* es defineixen els *EntityManager* i *DocumentManager*, eines de ORM (*Object-Relational Mapping*) i ODM (*Object-Document Mapping*) de *Symfony* que gestiona la persistència dels objectes del Model de forma transparent a la implementació física de la base de dades.

### 6.3 Gestió de la seguretat

*Symfony* proporciona eines de gestió de la seguretat<sup>18</sup> mitjançant *Firewalls*, ACL (*Access Control Lists*) i *Voters*. També proporciona eines de gestió d'usuaris i la creació de rols i relacions amb la resta d'objectes de la capa de *Model*.

Els ***Firewalls*** de *Symfony* actuen abans de l'execució d'un controlador i es configuren per especificar quines peticions (autenticades o anònimes) poden enviar-se cap a un controlador. En el nostre cas, hem configurat el *Firewall* per denegar tota petició anònima als controladors dels models de la nostra aplicació a excepció del controlador de la pàgina principal, que ha de permetre l'entrada a qualsevol usuari anònim pel simple fet de poder autenticar-se.

Una excepció especial del *Firewall* són les peticions externes realitzades a la REST API. Aquestes són peticions fetes a una adreça HTTP diferent

---

<sup>18</sup> <http://symfony.com/doc/current/book/security.html>

(<https://smartlab.caminstech.upc.edu/api/>) de l'aplicació i també tenen un controlador especial, com s'explicarà posteriorment.

```
firewalls:
  api:
    pattern: ^/api/
    security: false
  main:
    pattern: ^/
    form_login:
      check_path: /user/login_check
      login_path: /user/login
    logout:
      path: /user/logout
    anonymous: ~
    provider: caminstech_user_provider
```

*Codi 2: regles al Firewall de Symfony*

Per a l'autorització dels usuaris sobre les operacions a objectes del nostre Model fem ús dels **Voters**, classes del nostre *bundle* que s'executen abans d'executar una acció al *Controller* afectat i que decideixen per a un usuari autenticat i amb rols, quines accions poden realitzar sobre un objecte d'una *Entity* donada basant-se en la lògica de l'aplicació.

```
protected function voteOnAttribute($attribute, $entity, TokenInterface $token)
{
    $user = $token->getUser();
    if (!$user instanceof User) {
        return false;
    }
    switch($attribute) {
        case LaboratoryController::INDEX:
            return true;
        case LaboratoryController::VIEW:
            return ($entity->isManager($user) or $entity->isOperator($user));
        case LaboratoryController::UPDATE:
            return ($entity->isManager($user));
        case LaboratoryController::DELETE:
            return ($entity->isManager($user) and $entity->getExperiments()->count() == 0);
    }
    return false;
}
```

*Codi 3: voter de Laboratori*

Com es mostra al codi anterior, el *Voter* de *Laboratori* determina quines accions (INDEX, VIEW, UPDATE o DELETE) pot realitzar una instància d'Usuari (*User*) basant-se en els rols i les associacions que té amb aquest Laboratori.

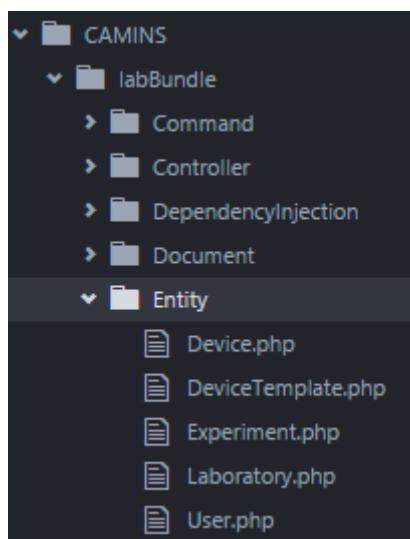
S'ha creat un *Voter* per cadascun de les Entitats del nostre *bundle*. En el cas que un usuari no tingui permisos per realitzar una acció sobre un *Entity* del *Model* es produeix una excepció que tractem en forma de missatge d'error o error HTTP en el cas de les APIs.

## 6.4 Entitats i Documents

Com s'ha mencionat al punt [6.2 – Arquitectura MVC](#), les Entitats (*Entity*) representen els models del món real. Al nostre sistema n'hem creat per Laboratori, Experiment, Dispositiu, Plantilla de Dispositiu i Usuari. Els atributs, relacions i cardinalitats es poden analitzar amb profunditat a l'apartat [4.3 – Model conceptual de Dades](#).

Cadascuna d'aquestes entitats reflecteix, mitjançant els seus atributs, les propietats d'aquell element al món real i que són d'interès pel nostre sistema. Així, per exemple, a l'entitat Laboratori disposem d'informació com el nom complet, acrònim, adreça, etc. però també especifiquem altres atributs com l'Usuari responsable (*Laboratory Manager*) que ens permeten definir les relacions entre entitats i que l'ORM interpretarà per crear l'esquema d'Entitat/Relació de la nostra base de dades.

La definició d'aquests atributs ha estat estudiada mitjançant els diagrames UML construïts prèviament a la implementació d'aquestes entitats.



*Il·lustració 49: entitats del nostre projecte*

```

class Device
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     * @JMS\Expose
     */
    private $id;

    /**
     * @var string
     *
     * @ORM\Column(name="fullname", type="string", length=255)
     * @JMS\Expose
     */
    private $fullname;

    /**
     * @var \DateTime
     *
     * @ORM\Column(name="createDateTime", type="datetime")
     * @JMS\Expose
     */
    private $createDateTime;

    /**
     * @var \DateTime
     *
     * @ORM\Column(name="update_time", type="datetime")
     * @JMS\Expose
     */
    private $updateTime;
}

```

*Il·lustració 50: part del codi de l'Entity Device*

A l'anterior il·lustració es pot identificar part d'una *Entity* amb alguns dels seus atributs i la seva visibilitat. Les anotacions posteriors formen part de l'ORM encarregat de *mapejar* els atributs de l'entitat en atributs d'una taula de la base de dades, com serà explicat més endavant. A l'*Entity* també es defineixen els atributs de relació amb altres entitats així com els *getters* i *setters* de tots els atributs.

Per altra banda, disposem d'un concepte similar a l'Entitat anomenat Document. Al nostre projecte disposem d'un tipus de dades que en diem "*Lectura d'un dispositiu*" (*Readings*) i és aquella dada provinent d'un dispositiu físic *Arduino* i que alhora recull un conjunt de sensors com poden ser temperatura, humitat, pressió, galga, etc. i una marca de temps. Aquesta dada es va estudiar com integrar-la dins de l'esquema conceptual del sistema, i es va observar que no tenia massa sentit realitzar cap operació sobre una dada en concret. Té més interès per l'usuari tractar un conjunt de dades per a poder realitzar posteriorment càlculs sobre elles. Per aquest motiu es va considerar que la màxima granularitat (o objecte més petit) al nostre esquema conceptual seria *Dispositiu*, i que aquest s'encarregaria de contenir, amb una altra estructura, el conjunt de dades



provinent d'aquest dispositiu físic. Aquest conjunt de dades és el que anomenem Document.

Un *Document* a *Symfony* utilitza un altre SGBD diferent dels relacionals convencionals. Es tracta de *MongoDB*, un SGBD NoSQL de tipus Document que encaixa perfectament amb el tipus de dades de sensors amb els qual treballem.

## 6.5 Base de dades relacional

Per emmagatzemar l'esquema relacional del nostre sistema web hem decidit escollir el SGBD *MySQL*<sup>19</sup> basant-nos en els següents criteris:

- **Base de dades de codi obert:** és molt popular en aplicacions web que corren en entorns LAMP. La seva popularitat ha fet que quasi qualsevol *framework* sigui compatible amb aquest SGBD. Compleix les expectatives sense necessitat de fer-nos requerir altres productes més cars com *Oracle* o *SQL Server*.
- **Simplicitat:** és fàcil d'aprendre, d'utilitzar i de trobar ajuda. En pocs passos disposem d'una base de dades plenament funcional i operativa.
- **Tecnologia provada:** es considera que *MySQL* és tecnologia provada per la indústria. Multitud d'aplicacions i sistemes l'utilitzen.
- **Suport transaccional:** *MySQL* és perfectament adient per a aplicacions transaccionals, com en el cas de la web. Satisfà les especificacions<sup>[10]</sup> ACID (Atomicitat, Consistència, Aïllament i Durabilitat) i integritat referencial d'aquestes.

Un altre SGBD útil que hagués satisfet aquestes propietats és *PostgreSQL*<sup>20</sup>, però degut a l'experiència tant per part dels desenvolupadors com l'equip d'operacions amb *MySQL* ens ha fet decantar per aquesta última.

## 6.6 Base de dades no relacional

Per emmagatzemar les lectures de dispositiu en forma de Document JSON utilitzem *MongoDB*<sup>21</sup>. Un SGBD no relacional i NoSQL de tipus *Document* el qual hem considerat oportú perquè satisfà els requisits que es desitja:

- **Propietats ACID menys crítiques:** una dada de sensor no requereix estrictament les propietats ACID (Atomicitat, Consistència, Aïllament i Durabilitat), que sí garanteix un SGBD relacional. Malgrat que no es garanteixi la consistència a

---

<sup>19</sup> <https://www.mysql.com/why-mysql/>

<sup>20</sup> <http://www.postgresql.org/>

<sup>21</sup> <https://www.mongodb.com/what-is-mongodb>

l'instant, no vol dir que *MongoDB* falli en aquesta propietat habitualment, (recordem que *MongoDB* és eventualment consistent) i en part també és important aquesta consistència per no perdre lectures de sensors i esbiaixar els estudis posteriors.

- **Format JSON de les dades provinents dels dispositius:** els dispositius *Arduino* programats als laboratoris ja proporcionen lectures de sensors en format JSON. Els Documents de *MongoDB* són estructures de dades basades en aquest mateix format. Sembla lògic doncs aprofitar aquest format per emmagatzemar-lo directament en un Document tal com arriba i sense cap tipus de tractament.
- **Dades no-relacionals:** les dades provinents dels dispositius *Arduino* (lectures de sensors) no estan relacionades amb cap altra Entitat ni Document més enllà de la mateixa entitat Dispositiu. *MongoDB* permet simular les relacions mitjançant estructures de subdocuments si escau.
- **Temporalitat de les dades:** es preveu que les lectures de sensor siguin d'utilitat durant un màxim de tres anys. L'escalabilitat de *MongoDB* és adequada pel volum d'informació estimat i la durabilitat d'aquestes.
- **Rendiment:** *MongoDB* té un rendiment adequat al tipus d'operacions més habituals que volem realitzar sobre les dades. És conegut que les bases de dades *NoSQL* tenen millor rendiment en les insercions i no pas en les modificacions. No realitzarem cap tipus de modificació de les dades un cop emmagatzemades, només operacions d'inserció, lectura i eliminació.

La decisió de mantenir l'esquema de la base de dades dividit en dos SGBD diferents s'ha pres amb cautela en motiu de la falta d'experiència en el camp de les bases de dades no-relacionals, no obstant això, el sistema avui dia funciona correctament. Per aquests motius, s'ha considerat[\[11\]](#) que la gestió d'aquest tipus de dades sigui portada a terme pel SGBD *MongoDB*.

Es necessita certa relació per identificar l'entitat *Dispositiu* (element del Model Relacional) amb el *Document* (element del Model no-Relacional) per representar el mateix concepte Dispositiu (recordem que un Dispositiu és una Entitat amb informació pròpia i recull el conjunt de lectures del dispositiu). Per aquest motiu, quan es crea, modifica i elimina una Entitat *Dispositiu* se *sincronitza* alhora el *Document* equivalent (Entitat Dispositiu i Document van sincronitzats, però només el Document conté les lectures dels sensors en JSON). Això dóna com a resultat un tractament *transparent* de l'entitat i dades de sensor a nivell d'aplicació.

```
{
  "_id" : ObjectId("56be4c6afb8b66d3047b23c6"),
  "readingData" : [
    {
      "jsondata" : "{\"temperature\": 4, \"maxtemp\":5, \"mintemp\":2}",
      "server_time" : ISODate("2016-02-12T21:21:28.000Z")
    },
    {
      "jsondata" : "{\"temperature\": 4, \"maxtemp\":5, \"mintemp\":2}",
      "server_time" : ISODate("2016-02-12T21:22:09.000Z")
    },
    {
      "jsondata" : "{\"temperature\":4,\"maxtemp\":5,\"mintemp\":2}",
      "server_time" : ISODate("2016-02-16T17:26:40.000Z")
    },
    {
      "jsondata" : "{\"temperature\":4,\"maxtemp\":5,\"mintemp\":2}",
      "server_time" : ISODate("2016-02-16T17:26:40.000Z")
    }
  ],
  "deviceId" : 3
}
```

*Codi 4: exemple de Document MongoDB per un Dispositiu "3"*

L'atribut *deviceId* identifica el *Document* a nivell d'aplicació i té el mateix valor que el *deviceId* de l'objecte de la entitat *Device*.

## 6.7 Serveis

Els Serveis (*Services*) van estretament lligats a la persistència i gestió de les dades. Tal com s'indica a la documentació i els exemples de *Symfony*, els controladors, per defecte, prenen la responsabilitat de crear, obtenir, modificar i eliminar les instàncies de les Entitats del nostre *bundle* mitjançant un altre servei del sistema anomenat *EntityManager* i que forma part del ORM/ODM de *Doctrine*.

A causa que diversos controladors han de fer certes tasques comunes amb les mateixes entitats s'ha decidit treure la responsabilitat de gestionar les dades als *Controllers* i traspassar-la a unes noves classes que hem anomenat Serveis (*Services*). En aquestes classes hem creat funcionalitats de creació, obtenció, actualització i eliminació de totes les entitats que s'han mencionat. S'ha creat un Servei per cadascuna de les Entitats del nostre *bundle* i també se n'ha creat per a la gestió de *Documents*.

```
class LaboratoryService
{
    private $doctrine;

    public function __construct($doctrine)
    {
        // ...

    public function getLaboratory($id)
    {
        $em = $this->doctrine->getManager();
        $laboratory = $em->getRepository('CAMINSlabBundle:Laboratory')->find($id);
        if (!$laboratory) {
            throw new NotFoundHttpException();
        }
        return $laboratory;
    }

    public function getLaboratories($user)
    {
        // ...

    public function createLaboratory()
    {
        // ...

    public function saveLaboratory(Laboratory $laboratory)
    {
        $em = $this->doctrine->getManager();
        if (!$em->contains($laboratory)) {
            $laboratory->setCreateTime(new \DateTime());
        }
        $laboratory->setUpdateTime(new \DateTime());
        $em->persist($laboratory);
        $em->flush();
    }

    public function deleteLaboratory($laboratory)
    {
        // ...
    }
}
```

Il·lustració 51: servei de Laboratori amb operacions de manipulació dels objectes del ORM

## 6.8 Controladors

Els Controladors (*Controllers*) són els encarregats de rebre i gestionar les peticions externes, tant d'usuaris, com d'altres aplicacions per mitjà d'una REST API. A *Symfony* els *Controllers* reben les peticions mitjançant l'assignació d'una URL i un METHOD HTTP a una *Action* del mateix *Controller*. Els hem classificat en dos tipus:

### 6.8.1 Controladors d'aplicació web

S'ha definit un Controlador (*Controller*) per cada una de les entitats que disposem. El Controlador d'una Entitat és l'encarregat de consultar al *Voter* si l'usuari que està realitzant la petició té permisos per realitzar-la; en tal cas, es posa en contacte amb el Servei d'entitat corresponent per realitzar l'operació sobre l'objecte sol·licitat (sigui creació, modificació, consulta o eliminació). Un cop gestionat l'objecte, el Controlador és responsable d'associar aquest objecte amb el seu Formulari (*Form*) i retornar-lo a la *Vista* per mostrar-lo a l'usuari mitjançant el sistema de plantilles *Twig*, que es detallarà més endavant.

```
class DeviceController extends Controller
{
    const INDEX = 'device_index';
    const VIEW = 'device_view';
    const DOWNLOAD = 'device_download';
    const CREATE = 'device_create';
    const UPDATE = 'device_update';
    const DELETE = 'device_delete';

    private $laboratoryService = null;
    private $experimentService = null;
    private $deviceService = null;
    private $jsonManager = null;

    public function init()
    {
        $this->laboratoryService = $this->get('laboratory_service');
        $this->experimentService = $this->get('experiment_service');
        $this->deviceService = $this->get('device_service');
        $this->jsonManager = $this->get('json_manager');
    }
}
```

*Il·lustració 52: controlador de Dispositiu instanciant els serveis amb els qual treballarà*

```

/**
 * Update an existing Experiment entity.
 *
 * @Route("/Laboratory/{laboratoryId}/experiment/{experimentId}/device/{id}/update", name="device_update")
 * @Method({"GET", "PUT"})
 */
public function updateAction(Request $request, $laboratoryId, $experimentId, $id)
{
    $this->init();
    $laboratory = $this->laboratoryService->getLaboratory($laboratoryId);
    $experiment = $this->experimentService->getExperiment($experimentId);
    $device = $this->deviceService->getDevice($id);
    if ($device->getExperiment()->getLaboratory() != $laboratory) {
        throw new NotFoundException();
    }
    $this->denyAccessUnlessGranted(self::UPDATE, $device);
    if ($template = $this->getFormTemplate($request, $device))
        return $this->render('CAMINSLabBundle:Device:update.html.twig', $template);
    return $this->redirectToRoute(self::INDEX, array('laboratoryId' => $laboratory->getId(), 'experimentId' => $experiment->getId()));
}

```

*Il·lustració 53: acció d'actualització d'un Experiment dins del Controlador d'Experiment*

## 6.8.2 Controladors de REST API

A la nostra aplicació es requereix a més d'un cas particular de *Controlador de Dispositiu* anomenat *DeviceRESTController* i que varia substancialment les funcionalitats de validació de les operacions sense *Voters*, així com una adaptació en retornar la informació a la capa de Vista. Aquest Controlador està dissenyat per operar com a API de l'aplicació web amb els dispositius *Arduino* mitjançant una RESTful API a través del protocol HTTP i els seus mètodes (GET, POST, PUT, DELETE).

El fet de treballar amb una API estàndard mitjançant el protocol HTTP permet que en un futur es pugui ampliar o modificar les possibilitats dels clients a altres plataformes més enllà d'*Arduino*.

A la nostra REST API s'empra JSON com a format d'intercanvi d'informació entre el client i el servidor. Això és així perquè els nostres dispositius *Arduino* ja treballaven amb JSON, fet que va propiciar decantar-nos clarament per aquest format d'intercanvi de dades més enllà del seu ús ja habitual en el món dels serveis web.

Per a la generació d'aquesta REST API ens hem ajudat d'un *bundle* extern anomenat *FOSRESTBundle*, i del *JMSSerializer* que ens ha permès convertir (o serialitzar) objectes de les nostres entitats a JSON i viceversa de manera automàtica.

```

class ReadingRestController extends VoryxController
{
    private $readingService = null;

    public function init()
    {
        $this->readingService = $this->get('reading_service');
    }

    /**
     * Update a ReadingDocument entity.
     *
     * @View(serializerEnableMaxDepthChecks=true)
     *
     * @param Request $request
     * @param $reading
     *
     * @return Response
     */
    public function putAction(Request $request, Reading $reading)
    {
        $this->init();
        try {
            $providedData = $this->get('request')->request->get('data');
            $providedToken = $this->get('request')->request->get('token');
            if ($reading->getToken() == $providedToken) {
                $this->readingService->saveReadingData($reading,$providedData);
                return $reading;
            } else {
                return FOSView::create("Wrong or no token provided.", Codes::HTTP_UNAUTHORIZED);
            }
        } catch (\Exception $e) {
            return FOSView::create("Internal server error.", Codes::HTTP_INTERNAL_SERVER_ERROR);
        }
    }
}

```

Il·lustració 54: controlador API de Dispositiu

## 6.9 Capa de Vista

A *Symfony*, i a qualsevol projecte i *framework* web, les vistes juguen un paper fonamental a l'hora de presentar la informació a l'usuari.

A aplicacions web antigues no hi havia una bona separació entre el codi de la lògica de negoci i el codi encarregat de mostrar elements visuals HTML resultants. Per aquest motiu van aparèixer els *Template Engines*.

Un *Template Engine* és un component nou amb l'objectiu de crear aquesta separació entre lògica i presentació d'una aplicació. Això, a part de flexibilitat a l'hora de poder dissenyar ambdues parts independentment, ens proporciona seguretat afegida, ja que aquests motors inclouen eines de neteja de *strings* i eviten possibles vulnerabilitats conegudes al món web.

A *Symfony* aquest *Template Engine* s'anomena *Twig* i té, ahora, el seu propi llenguatge de marques que s'ubica al ja conegut llenguatge de marques HTML i CSS per transmetre la informació des de la capa del *Controlador* a la capa de *Vista*.

A continuació es mostra un exemple de codi *Twig* (reconegut pels blocs `{% %}`) intercalat a l'HTML, emprat per mostrar informació dels Laboratoris.

```
{% for laboratory in laboratories['entities'] %}
<tr class="laboratory">
  <td class="laboratory-acronym">{{ laboratory.acronym }}</td>
  <td class="laboratory-fullname">{{ laboratory.fullname }}</td>
  <td class="laboratory-experiments" style="text-align:center">{{ laboratory.experiments.count }}</td>
  <td class="laboratory-manager">{{ laboratory.manager.username }}</td>
  <td>
    {% if is_granted('experiment_index', laboratory) %}
      <a class="btn btn-primary btn-sm" href="{{ path('experiment_index', { 'laboratoryId': laboratory.id }) }}">Experiments ({{laboratory.experiments.count}})</a>
    {% endif %}
    {% if is_granted('experiment_index', laboratory) %}
      <a class="btn btn-default btn-sm" href="{{ path('deviceTemplate_index',{'laboratoryId': laboratory.id}) }}">Dev. Templates ({{laboratory.deviceTemplates.count}})</a>
    {% endif %}
    {% if is_granted('laboratory_view', laboratory) %}
      <a class="btn btn-default btn-sm" href="{{ path('laboratory_view', { 'id': laboratory.id }) }}">Details</a>
    {% endif %}
    {% if is_granted('laboratory_update', laboratory) %}
      <a class="btn btn-default btn-sm" href="{{ path('laboratory_update', { 'id': laboratory.id }) }}">Update</a>
    {% endif %}
    {% if is_granted('laboratory_delete', laboratory) %}
      <a class="btn btn-danger btn-sm" href="{{ path('laboratory_delete', { 'id': laboratory.id }) }}">Delete</a>
    {% endif %}
  </td>
</tr>
{% endfor %}
```

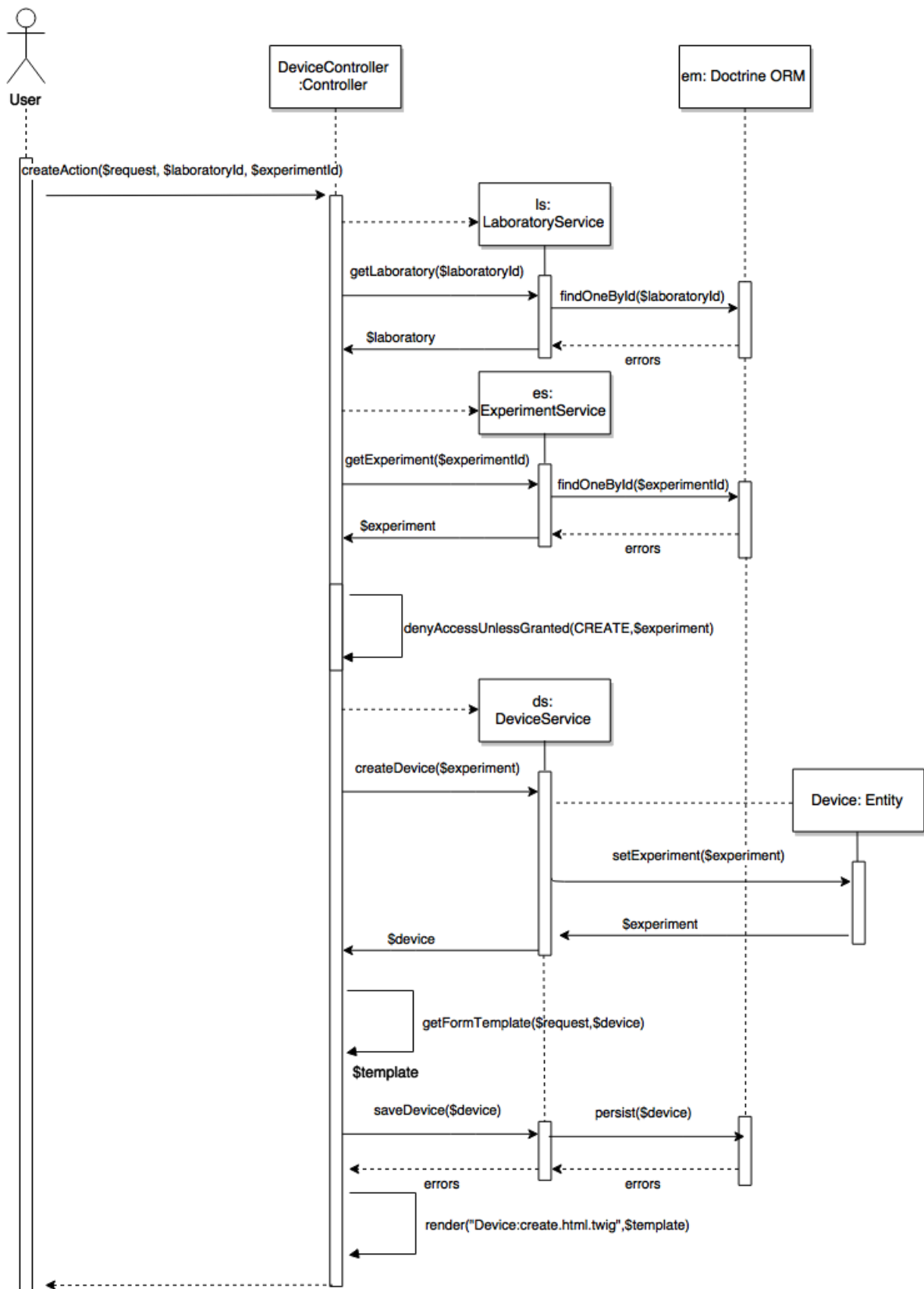
*Il·lustració 55: els blocs entre {} és codi Twig i s'encarrega d'incrustar a l'HTML les dades enviades pel controlador*

## 6.10 Diagrames de Seqüència

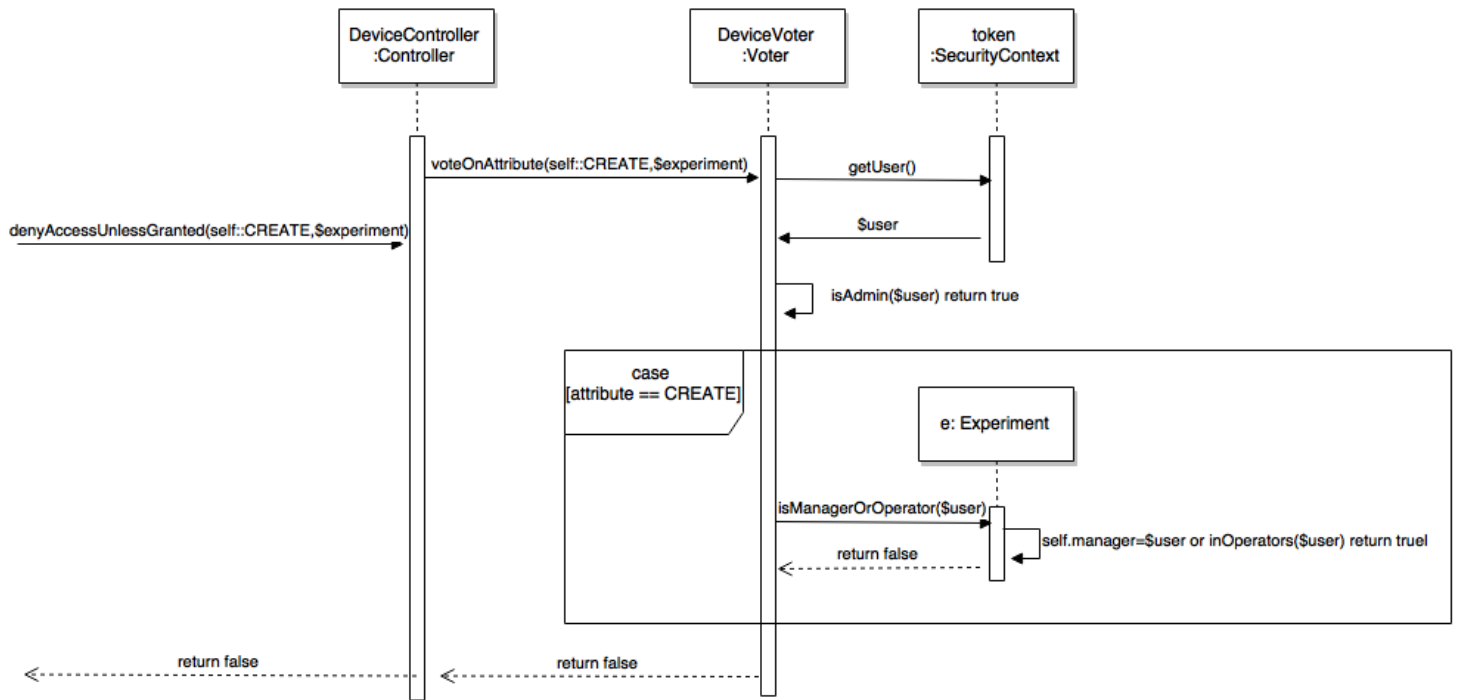
Per entendre millor la interacció entre els diferents components explicats que prenen part en el flux d'execució d'una funcionalitat, tot seguit s'inclouen els diagrames de seqüència del cas d'ús [4.1.5.2 – Creació d'un Dispositiu](#) en el qual entren en joc els components bàsics descrits del sistema, així com els de la publicació d'una Lectura de Dispositiu mitjançant la REST API del sistema.



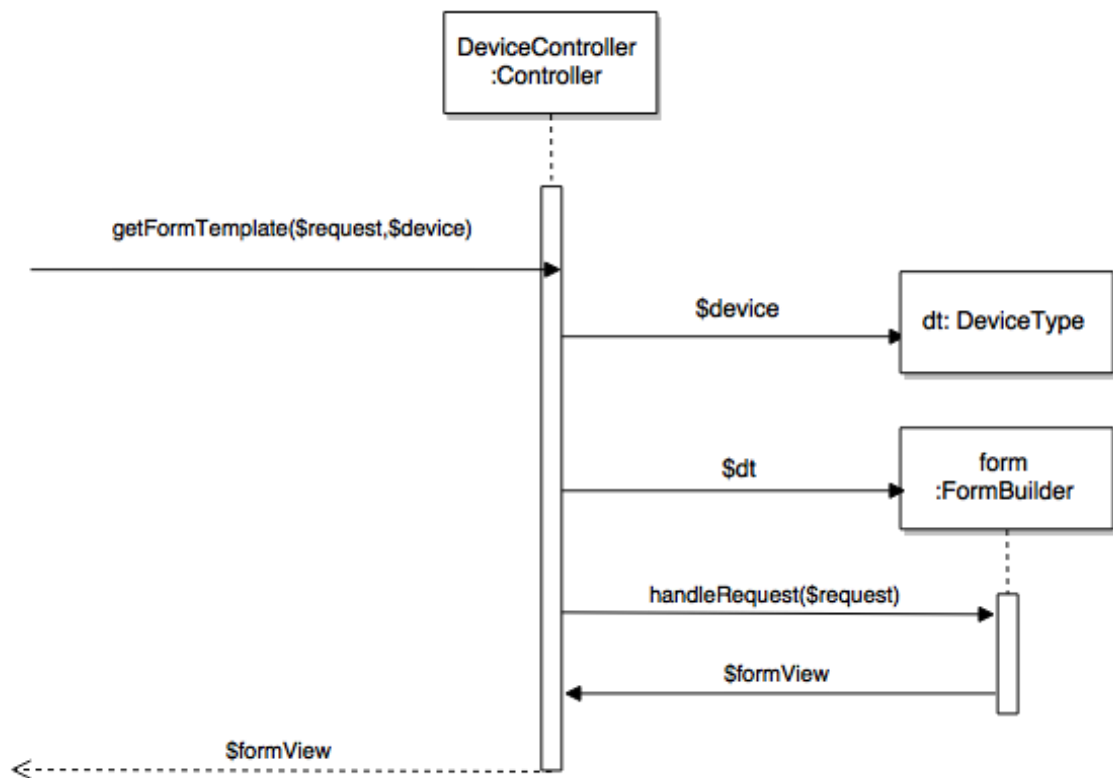
### 6.10.1 Diagrama de seqüència Creació d'un Dispositiu



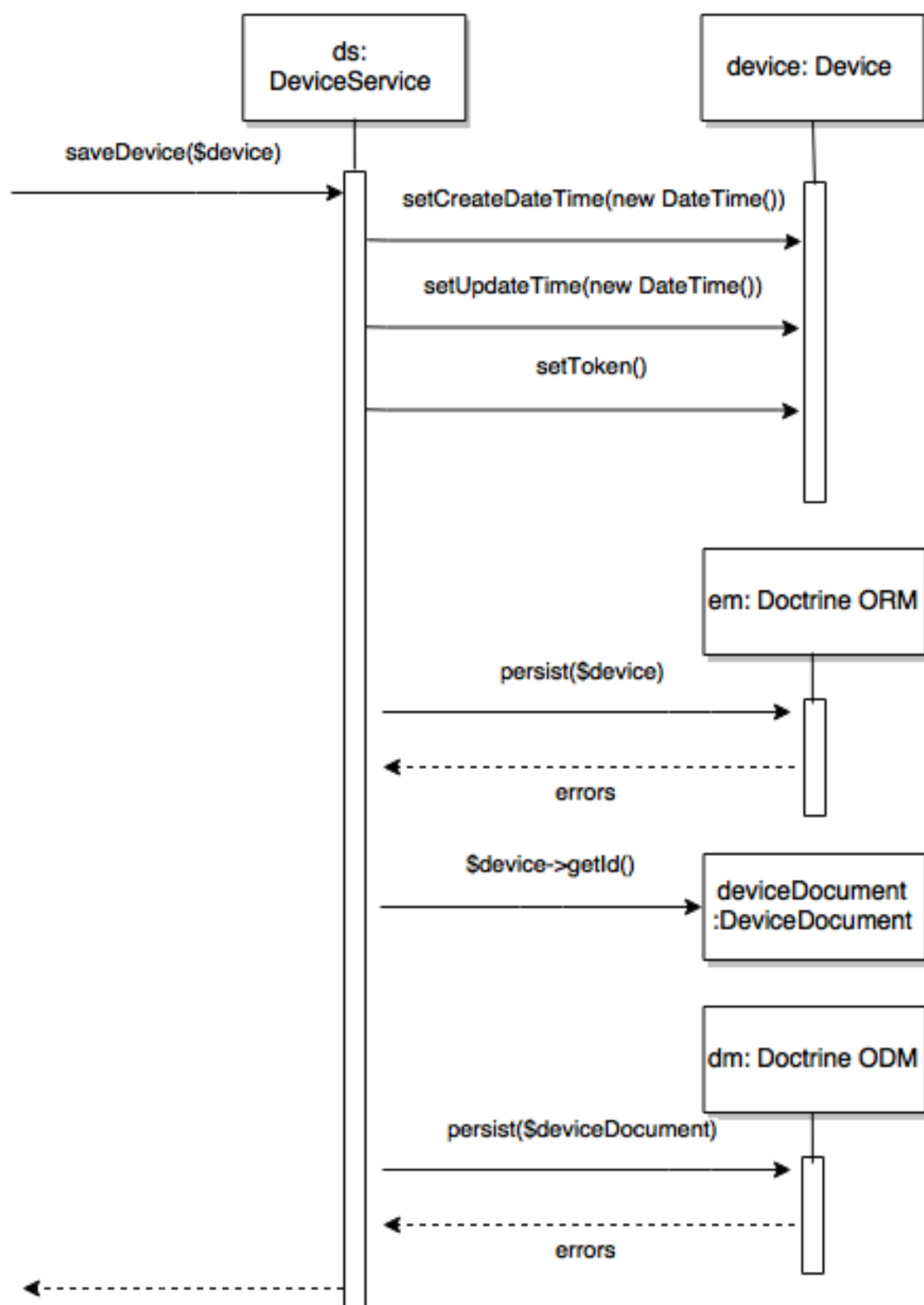
Il·lustració 56: diagrama de seqüència general del CU: Creació d'un Dispositiu d'Experiment de Laboratori



Il·lustració 58: validació dels permisos de l'usuari sobre la creació d'un Dispositiu en un Experiment



Il·lustració 57: generació del Formulari i validació de dades d'aquest

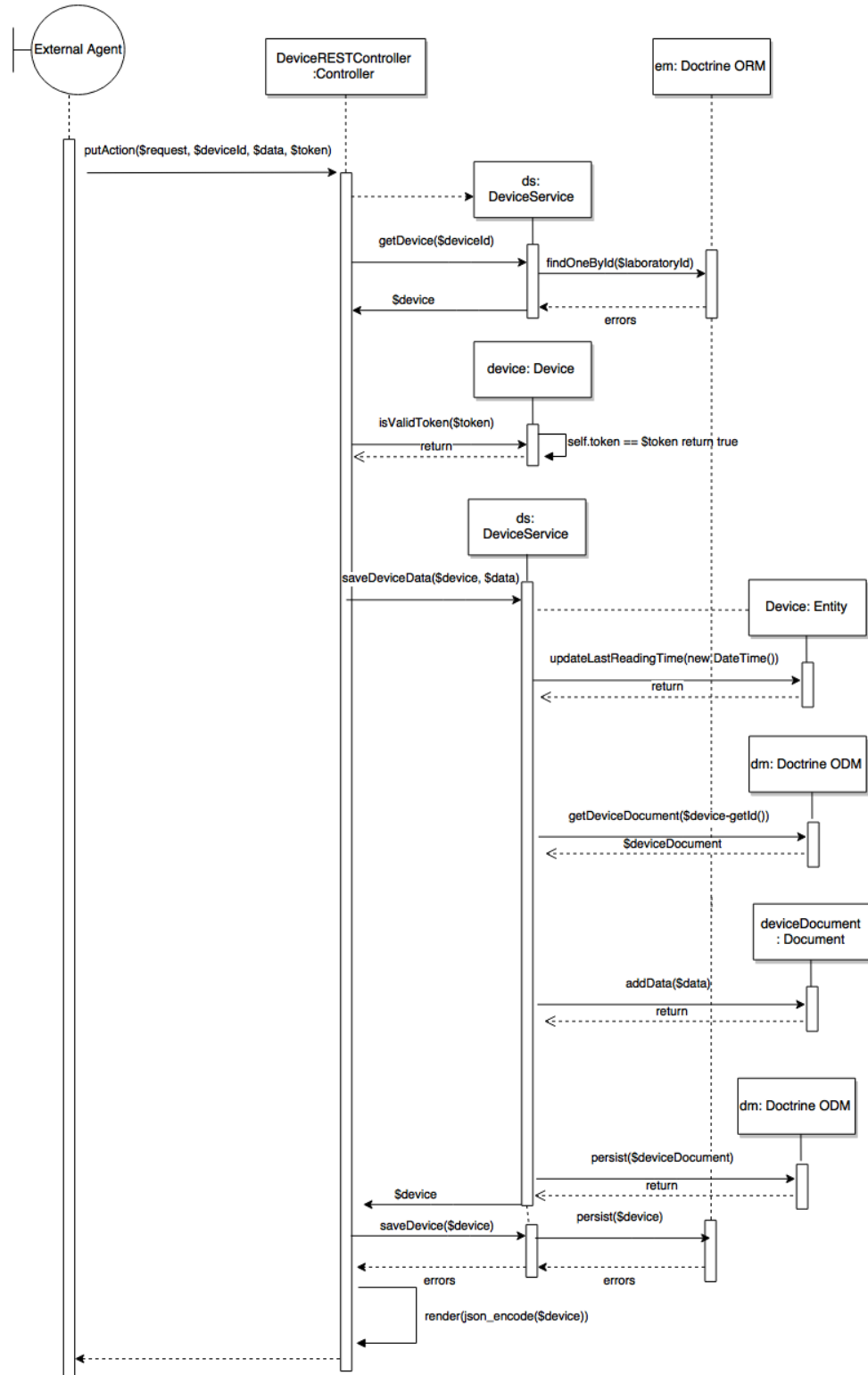


Il·lustració 59: persistència de l'entitat Device i del document "deviceDocument"

L'elecció d'aquest cas d'ús ve donat perquè es fa ús dels "Servei de Laboratori" i "Servei d'Experiment" per obtenir els objectes, i la creació d'un dispositiu dins d'un experiment d'un laboratori. El patró és el mateix a la creació d'un experiment dins d'un laboratori i la creació d'un laboratori, així com la resta d'operacions d'un Controlador, que són molt

semblants a les operacions de creació, edició, visualització i eliminació. A més, aquest cas d'ús té la peculiaritat de treballar també amb el *DeviceDocument* i el *Doctrine ODM*, fet que el fa més interessant encara.

#### 6.10.2 Diagrama de seqüència de l'acció PUT lectura de dispositiu a la REST API.



Il·lustració 60: diagrama de seqüència de la interacció amb la REST API

## 6.11 Descripció de la RESTful API

La RESTful API del sistema proporciona un punt d'entrada al sistema per a agents externs. En el nostre cas, proporciona als dispositius *Arduino* la capacitat de publicar mitjançant el protocol HTTP lectures de sensors al sistema web.

### 6.11.1 Publicació d'una lectura de sensors d'un Dispositiu

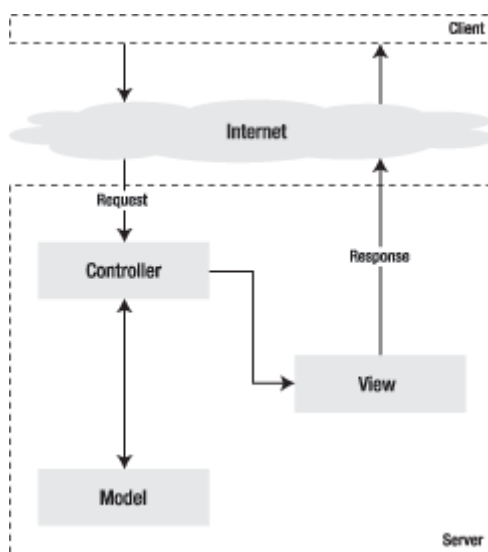
El comportament estàndard d'una lectura de dispositiu requereix que el dispositiu que publica aquesta lectura estigui donat d'alta al sistema en un experiment d'un laboratori. En donar d'alta un dispositiu al sistema s'assigna a aquest un identificador de dispositiu (*idDevice*) i un *token* únic per aquest dispositiu i que haurà de ser adjuntat a la requesta REST. La petició a la REST API ha de satisfer les propietats següents:

Mètode PUT de Device		
Adreça URL del recurs	https://smartlab.caminstech.upc.edu/api/devices/{ <i>idDevice</i> }	
Informació sobre el recurs	Format de resposta: JSON	
Paràmetres (Obligatori)	<b>Clau</b>	<b>Tipus</b>
	<i>data</i>	<i>jsonObject</i>
	<i>token</i>	<i>token</i>
Requesta d'exemple	<pre>{   data:{ "temperature":"15",          "humidity":"76",          "dateTime":"2016-03-15 20:28"         },   token:"eqwkj2342jkkj23jkdfkjsf21123" }</pre>	
Resposta d'exemple	<pre>{   "id":1,   "fullname":"<b>Arduinol</b>",   "create_date_time":"2016-03-17T14:36:39+0000",   "update_time":"2016-04-06T17:09:45+0000" }</pre>	
Codis d'error	403 <i>Forbidden. Experiment closed.</i> 401 <i>Unauthorized. Invalid deviceId or token provided.</i>	
Codi de confirmació	201 <i>Created.</i>	

## 6.12 Patrons de disseny

Molts dels components *software* de *Symfony* emprats estan implementats fent ús de patrons de disseny[12] orientats a objecte, evidenciant la importància de crear components amb baix acoblament. En aquest capítol en descriurem els més rellevants, alguns d'ells no els hem implementat pròpiament a la nostra aplicació però si n'hem fet ús indirectament mitjançant la instanciació d'aquests components.

Malgrat que hem fet referència a què *Symfony* és un *framework* MVC<sup>22</sup>, hem de remarcar que, prèviament, qualsevol aplicació web segueix inicialment el patró *Request - Response*, ja que aquest tipus d'aplicacions no tenen estat (*stateless*) tal com sí que en tenen les aplicacions d'escriptori. Per sobre d'aquesta capa sí que s'aplica el patró MVC.



Il·lustració 61: patró MVC sobre el patró Request-Response

A continuació es descriuen els patrons de disseny que s'han utilitzat i en quins casos s'han fet servir o identificat:

- **Request – Response:** com s'ha mencionat, qualsevol aplicació web segueix primerament aquest patró d'intercanvi de missatges en el qual un client realitza una requesta a un servidor web, i aquest retorna una resposta (*response*). Típic a l'arquitectura client-servidor.
- **MVC:** l'arquitectura MVC separa la lògica de negoci (Model) de la presentació (Vista). El Controlador oculta el protocol utilitzat per fer la requesta (HTTP, consola, correu, etc.) del model i de la vista. I el Model genera una abstracció de la gestió de les dades, el qual fa que la vista i l'acció siguin independents, en aquest cas, del tipus de base de dades emprat per la nostra aplicació. *Symfony* fa ús plenament d'aquesta arquitectura separant els components en les capes mencionades.

<sup>22</sup> [http://symfony.com/legacy/doc/gentle-introduction/1\\_4/en/02-Exploring-Symfony-s-Code](http://symfony.com/legacy/doc/gentle-introduction/1_4/en/02-Exploring-Symfony-s-Code)

- **Front Controller:** un controlador que gestiona totes les requestes realitzades a un lloc web. En el cas de *Symfony* es tracta del *web/app.php* des d'on es gestionen totes les requestes cap a la resta de controladors de la nostra aplicació.

```
$kernel = new AppKernel('prod', false);
$kernel->loadClassCache();
$kernel = new AppCache($kernel);
$request = Request::createFromGlobals();
$response = $kernel->handle($request);
$response->send();
$kernel->terminate($request, $response);
```

Il·lustració 62: part del FrontController de *Symfony*

- **Mediator Pattern:** patró que defineix un objecte que especifica com un seguit d'altres objectes interactuen entre ells. Promou el baix acoblament evitant que els objectes es referencïin entre ells explícitament i permet, per tant, alterar la seva interacció independentment. És el cas del *EventDispatcher* de *Symfony*, que permet als diferents components de la nostra aplicació comunicar-se entre elles mitjançant l'execució i escolta d'events. Alguns exemples de components que segueixen aquest comportament són *RouterListener* i *Firewall*.
- **Dependency Injection:** patró que permet l'eliminació de dependències de codi amb funcionalitats habituals transformant-los en components estàndards plenament funcionals per sí mateixos, fet que permet el seu ús o canvi en temps d'execució o compilació. És el cas de tots els mòduls de *Symfony* com *Doctrine* (ORM), motors de correu (*Mailer*), motors de renderització (*Twig*), etc. així com tot el conjunt de serveis propis definits (*LaboratoryService*, *ExperimentService*, *ReadingService*, *UserService*) que són instanciats com contenidors independents.
- **Factory Pattern:** patró que defineix una interfície per crear un objecte però delega a les subclasses la responsabilitat d'escollir quina classe instanciar. És el cas de la creació d'usuaris, que segons el rol assignat, es creen diferents tipus d'objecte *User* amb diferents responsabilitats i atributs.
- **Form Builder Interface:** patró que separa la construcció d'objectes complexos de la seva representació, de tal manera que el procés de construcció pugui crear diferents representacions. És el cas del *FormBuilderInterface* de *Symfony* que ens permet definir els camps i atributs del formulari HTML en el procés de la construcció d'aquest objecte.

```
$form = $this->createForm($entityType, $entity, array('action' => $action, 'attr' => array('class' => 'form-edit'), 'method' => 'POST'));
$form->add('submit', 'submit', array('label' => 'Create', 'attr' => array('class' => 'btn btn-default')));
```

Il·lustració 63: creació del formulari HTML de *Laboratori* amb mètode *POST*

- **Unit of work:** patró utilitzat per l'ORM *Doctrine* que té constància dels canvis que pateix un objecte durant les transaccions i efectua els canvis a la base de dades de forma ordenada. N'hem fet ús extensament quan després de modificar un objecte, per exemple, en eliminacions en cascada d'objectes, el ORM s'encarrega d'executar tot el seguit d'operacions SQL per realitzar els canvis oportuns a la base de dades.

```
public function deletelaboratory($laboratory)
{
    $em = $this->doctrine->getManager();
    $em->remove($laboratory);
    $em->flush();
}
```

*Il·lustració 64: l'operació \$em->flush() elimina el Laboratori i tots els seus objectes relacionats en cascada, realitzant els canvis oportuns a la BDD*

- **Proxy:** patró que permet substituir atributs o components d'un altre objecte per tal de controlar l'accés a aquests. No n'hem fet ús explícitament però sí que el *Doctrine* en fa ús quan intentem accedir a atributs d'objectes que no hem carregat de la base de dades creant un *Proxy* de l'objecte sol·licitat i permetent la càrrega d'aquests atributs posteriorment mitjançant el *Lazy Loading*.



## 6.13 Tests

Per garantir un nivell de qualitat de les funcionalitats de la nostra aplicació és necessari la validació d'aquestes mitjançant l'ús de Tests. *Symfony*, mitjançant el motor *PHPunit*<sup>23</sup>, permet realitzar tests unitaris i funcionals.

A la nostra aplicació s'ha considerat necessari realitzar tests funcionals, que són aquells que validen la integració de diferents components de la nostra aplicació (des de l'enrutament fins a les vistes). Un Test funcional a *Symfony* realitza el següent procés d'execució:

1. Realitza una petició HTTP.
2. Valida la resposta de la petició.
3. Segueix un enllaç, envia un formulari o segueix una redirecció del servidor.
4. Comprova la resposta del servidor.

Per la nostra aplicació, hem realitzat un test per cadascun dels Controladors i cadascuna de les accions d'aquests per tal de garantir el correcte funcionament de les funcionalitats descrites a [4.1 – Casos d'Ús](#).

Algunes operacions dels tests necessiten dades prèvies per funcionar. Per poder validar tots els casos d'ús d'aquests controladors es necessita disposar de dades d'usuaris, laboratoris, experiments o dispositius en cada cas. Aquest conjunt de dades temporals s'anomenen *Fixtures* i es creen i destrueixen a cada execució del test.

Els nostres *Tests* estan estructurats en tres components:

- ***WebTestCaseCommon***: classe que inclou operacions comunes a tots els tests sobre la validació de funcionalitats, l'autenticació, respostes i redireccions del servidor, i comparació dels objectes DOM resultants amb els esperats.
- ***Entities***: classe amb les operacions d'inserció, modificació i eliminació de les dades temporals o *Fixtures*.
- ***LaboratoryControllerTest***: classe *Test* del controlador *LaboratoryController* el qual realitza l'execució de totes les funcionalitats d'aquest i realitza la validació de l'estat del sistema al finalitzar l'execució comparant l'estat esperat amb el real.

A continuació s'explica el *Test* sobre les funcionalitats de *LaboratoryController*:

L'objectiu del *WebTestCaseCommon* és encapsular totes les operacions comunes als tests com la inicialització dels *Fixtures*, el procés d'autenticació i comparació de la

---

<sup>23</sup> <http://symfony.com/doc/current/book/testing.html>

resposta del servidor amb l'esperada en el test. A la següent figura es poden observar aquestes funcions.

```
protected function setFixtures($fixtures)
{
    $this->fixtures = $fixtures;
}

protected function authenticate($username)
{
    $user = $this->entitites->getUserByUsername($username);
    $crawler = $this->client->request('GET', '/user/login');
    $form = $crawler->selectButton('Log In')->form();
    $crawler = $this->client->submit($form, array('_username' => $user->getUsername(), '_password' => $user->getUsername()));
    return $user;
}

protected function assertCurrentUrl($desired, $followRedirect = false)
{
    if ($followRedirect) {
        $this->client->followRedirect();
    }
    $current = parse_url($this->client->getRequest()->getUri(), PHP_URL_PATH);
    $this->assertEquals($desired, $current, 'Failed asserting that two urls are equal.');
```

```
protected function assertHttpSuccess($followRedirect = false)
{
    if ($followRedirect) {
        $this->client->followRedirect();
    }
    $this->assertEquals(Response::HTTP_OK, $this->client->getResponse()->getStatusCode());
}

protected function assertHttpForbidden()
{
    $this->assertEquals(Response::HTTP_FORBIDDEN, $this->client->getResponse()->getStatusCode());
}
```

*II-lustració 65: funcions d'autenticació i validació de redireccions i HTTP Response*

Una funció destacable d'aquest component és el «*assertTableEquals*» que permet comparar els objectes DOM resultants amb els esperats per l'operació executada al test. Així doncs, si s'ha creat un objecte *Laboratory* nou, l'objecte DOM resultant haurà de reflectir aquest canvi contenint les dades del laboratori.

```
protected function assertTableEquals($expecteds, $crawler, $equals)
{
    $actuals = $crawler->each(function($str) { return $str; });

    $this->assertEquals(count($expecteds), count($actuals));
    foreach($expecteds as $expected) {
        $found = false;
        foreach ($actuals as $actual) {
            $found = $found || $equals($expected, $actual);
        }
        $this->assertTrue($found, 'Element "'.(string)$expected.'" not found.');
```

```
    }
    foreach($actuals as $actual) {
        $found = false;
        foreach ($expecteds as $expected) {
            $found = $found || $equals($expected, $actual);
        }
        $value = implode('|', array_filter(explode('|', preg_replace('/\n +/', '|', $actual->text()))));
        $this->assertTrue($found, 'Element "'.$value.'" not expected.');
```

```
    }
}
```

*II-lustració 66: funció «assertTableEquals()» que verifica la comparació dels objectes DOM resultants amb els esperats per l'operació del test*

El següent component dels nostres tests és l'anomenat *Entity*. Aquest component és l'encarregat de proporcionar la gestió de les dades temporals o *Fixtures* per a l'execució dels Tests.

```
public function __construct(ObjectManager $objectManager)
{
    $this->userManager = $objectManager->getRepository(self::ENTITY_USER);
    $this->laboratoryManager = $objectManager->getRepository(self::ENTITY_LABORATORY);
}

public function getUser($id)
{
    return $this->userManager->find($id);
}

public function getUserByUsername($username)
{
    return $this->userManager->findOneByUsername($username);
}

public function getUsers()
{
    return $this->userManager->findAll();
}

public function getUsersByAdmin($isAdmin)
{
    return array_filter($this->getUsers(), function($o) use ($isAdmin) { return $isAdmin ? $o->isAdmin() : !$o->isAdmin(); });
}

public function getLaboratory($id)
{
    return $this->laboratoryManager->find($id);
}
```

*Il·lustració 67: algunes de les operacions de manipulació dels Fixtures.*

Finalment, el tercer component que completa el Test és, en aquest cas, el *LaboratoryControllerTest* i que conté totes les operacions a efectuar sobre el *LaboratoryController* per validar el seu comportament. El mateix procediment s'aplica en els Tests dels altres controladors.

```
class LaboratoryControllerTest extends WebTestCaseCommon
{
    public function setUp()
    {
        $this->setFixtures([new FixtureSet1]);
        parent::setUp();
    }
}
```

*Il·lustració 69: el Test carrega el seu conjunt de dades necessari per l'execució "FitxtureSet1"*

```
public function testCreate()
{
    $laboratory = FixtureCommon::createLaboratory('lab-new', $this->entitites->getUserByUsername('user1'), [ $this->entitites->getUserByUsername('user2') ]);

    $this->authenticate('admin');
    $crawler = $this->client->request('GET', '/laboratory/create');
    $this->assertHttpSuccess();

    $this->submitLaboratoryForm($crawler, $laboratory);
    $this->assertHttpSuccess($followRedirect = true);

    $found = $this->entitites->getLaboratoryByAcronym($laboratory->getAcronym());
    $this->assertNotNull($found);
    $this->assertLaboratoryEquals($laboratory, $found);
}
```

*Il·lustració 68: operació que testeja de la creació d'un Laboratory*

Com es pot observar a la figura anterior, l'operació «*testCreate*» realitza l'autenticació amb un rol d'administrador i redirigeix el *crawler* a la pàgina de creació d'un laboratori. Seguidament es comprova que la resposta HTTP del servidor és correcta i procedeix a la creació del *Laboratory* mitjançant l'enviament del formulari i la comprovació un altre cop de la resposta del servidor. Finalment verifica que el laboratori s'ha creat buscant l'objecte DOM a la taula HTML de Laboratoris.

```
private function assertLaboratoryTableEquals($laboratories, $crawler)
{
    $this->assertTableEquals($laboratories, $crawler->filter('table > tbody > tr.laboratory'), function($user, $node) {
        return $user->getAcronym() == $node->filter('td.laboratory-acronym')->text() &&
            $user->getFullName() == $node->filter('td.laboratory-fullname')->text() &&
            count($user->getExperiments()) == $node->filter('td.laboratory-experiments')->text() &&
            $user->getManager()->getUsername() == $node->filter('td.laboratory-manager')->text();
    });
}

private function assertLaboratoryEquals($expected, $actual)
{
    $this->assertEquals($expected->getFullName(), $actual->getFullName());
    $this->assertEquals($expected->getAcronym(), $actual->getAcronym());
    $this->assertEquals($expected->getSpecialization(), $actual->getSpecialization());
    $this->assertEquals($expected->getAddress(), $actual->getAddress());
    $this->assertEquals($expected->getDescription(), $actual->getDescription());
    $this->assertOneRelationEquals($expected->getManager(), $actual->getManager());
    $this->assertManyRelationEquals($expected->getOperators(), $actual->getOperators());
}
```

*Il·lustració 70: operacions de comparació dels objectes DOM resultants amb els esperats per Laboratoris.*

El mateix procediment s'aplica per la resta d'operacions del Controlador, així com el dels altres controladors del nostre sistema.

Aquest test s'executa des del CLI mitjançant el programa **PHPunit** inclòs com a *Vendor* de *Symfony*.

## 7. Tecnologies emprades a la implementació

---

En aquest apartat es descriuran les tecnologies emprades i es posen exemples de les d'algunes implementacions destacables del projecte.

### 7.1 Symfony

*Symfony* és un *framework* web de codi obert basat en llenguatge PHP per a la creació d'aplicacions MVC de manera ràpida i escalable. L'elecció d'aquest *framework* està motivada per diversos factors:

- S'ajusta correctament a l'entorn de *DevOps* de *CaminsTech*.
- Té un bon rendiment.
- *Symfony* està orientat a construir aplicacions robustes en entorns empresarials i alhora dóna un control absolut de la seva configuració.
- Disposa d'una gran facilitat de creació i compartició de llibreries o *bundles* que agilitzen la integració de funcionalitats habituals (gestió d'usuaris, creació d'una API REST, etc.) en altres projectes, tot permetent el reaprofitament de codi.
- Inclou mecanismes de seguretat comuns per evitar fallades de seguretat[\[13\]](#) en aplicacions web (CSRF, *SQL Injection*, XSS etc.).

Inclou eines molt potents de depuració del codi i creació de tests i documentació.

### 7.2 Eines de DevOps

*Symfony* s'integra correctament amb les eines *DevOps* que s'utilitzen a *CaminsTech*. *DevOps* implica una manera de treballar en equip estretament lligada entre l'equip de desenvolupadors i l'equip de sistemes d'una organització. Fins a l'aparició d'aquestes eines, era habitual que un grup de desenvolupadors hagués de mantenir un entorn d'execució comú (*dev*, *testing*, *production*) o bé construir-se'n el seu en local. Això provocava dificultats a l'hora de compartir un mateix codi entre entorns d'execució dels diferents programadors del grup de treball a causa del fet que la configuració de l'entorn d'execució no sempre era igual per tots ells.

Amb *DevOps*, i concretament a *CaminsTech* es fa ús de *Vagrant*<sup>24</sup>. Aquestes eines permeten definir les característiques d'un entorn d'execució (una màquina virtual i tot el seguit d'aplicacions de servidor com servidor web, base de dades, etc. necessàries per executar l'aplicació web *Symfony*) mitjançant fitxers de text parametritzables. És el que es coneix com *Infrastructure As Code*<sup>25</sup>, el concepte de definir recursos informàtics i

---

<sup>24</sup> <https://www.vagrantup.com/>

<sup>25</sup> <http://martinfowler.com/bliki/InfrastructureAsCode.html>

d'infraestructura mitjançant codi que es pot tractar com qualsevol altre sistema software.

L'*Infrastructure As Code* es basa en les següents pràctiques:

- Ús de fitxers de definició de configuracions.
- Control de versions d'aquestes definicions.
- Tests continus de sistemes i processos.
- Alta disponibilitat dels serveis mitjançant canvis incrementals.

Aquestes eines ens aporten avantatge a la infraestructura dinàmica, mitjançant la creació i actualització fàcil de servidors quan es necessiten noves configuracions o la càrrega de treball disminueix.

### 7.3 Servidors

- **Cloud UPC:** és el servei d'equips virtuals de la UPC i que ens allotja el *CaminsSmartLab* dins de l'entorn segur d'aplicacions de la UPC. El manteniment i la gestió d'aquest equip és dut per l'equip d'operacions de *CaminsTech* i per tant recau fora de l'abast d'aquest projecte.
- **Apache** és el servidor web emprat dins de l'entorn LAMP definit per *Vagrant*. Ens ha permès definir un *Virtualhost* pel nostre domini (<https://smartlab.caminstech.upc.edu>) i especificar un certificat per acceptar connexions SSL mitjançant HTTPS.
- **MongoDB:** és el servidor de base de dades no relacionals (NoSQL) emprat per emmagatzemar dades de sensors. La seva instal·lació i configuració s'ha definit al *Vagrant*.
- **MySQL:** és el servidor de base de dades relacionals emprat per emmagatzemar l'esquema de l'aplicació. La seva instal·lació i configuració s'ha definit al *Vagrant*.
- **Client-side web Technologies:** per la capa de la Vista de l'aplicació s'han utilitzat elements d'HTML, CSS, *Javascript* i *framework* de client tals com *jQuery*<sup>26</sup> i *Bootstrap*<sup>27</sup> per millorar l'experiència de l'usuari.

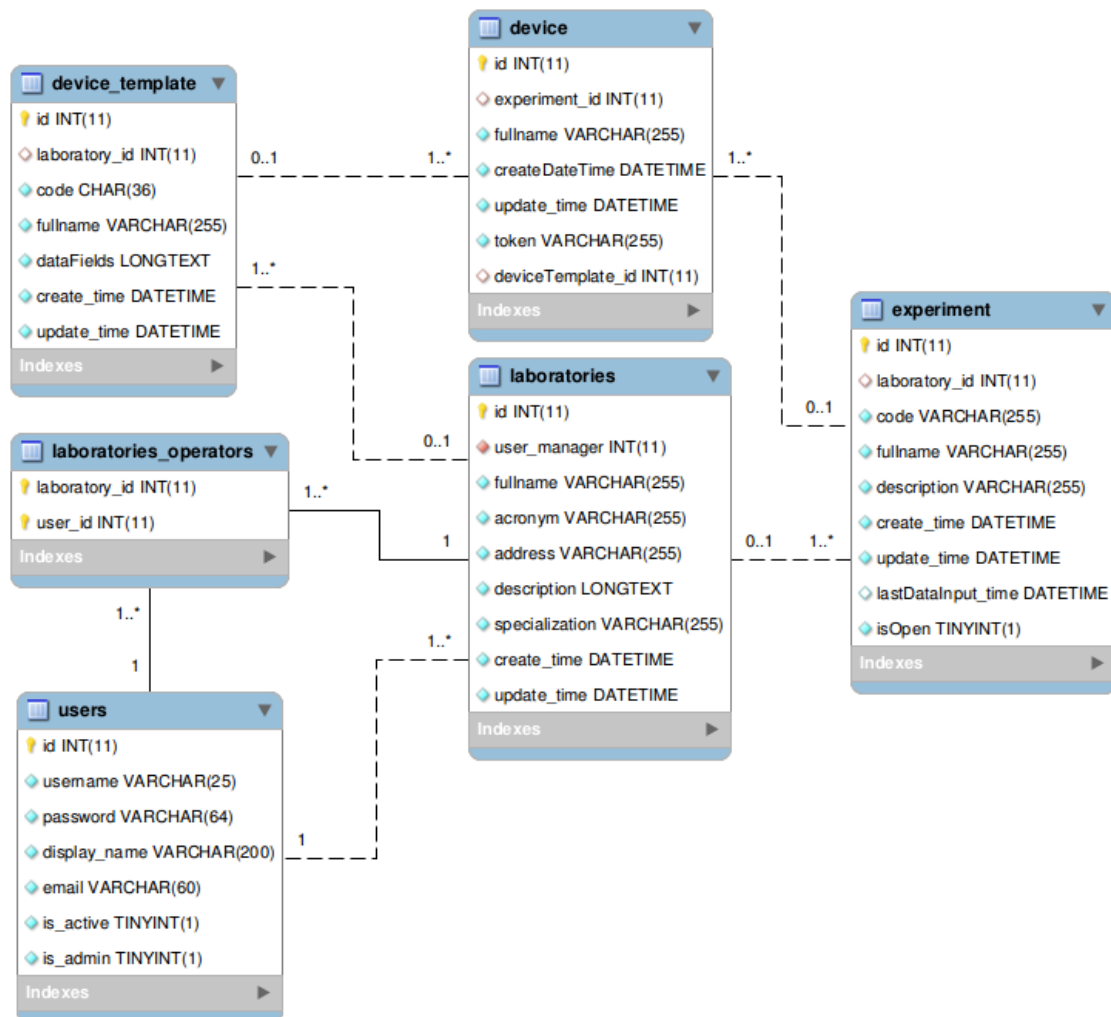
---

<sup>26</sup> <https://jquery.com/>

<sup>27</sup> <http://getbootstrap.com/>

## 7.4 Eines de persistència i abstracció de base de dades

L'esquema resultant de la definició del nostre *Model de Dades* i la seva traducció a Entitat/Relació és el següent:



Il·lustració 71: esquema relacional de la base de dades MySQL

Per definir aquest esquema Entitat/Relació ens hem ajudat de les eines d'ORM *Doctrine* de *Symfony*. Aquest ORM ens permet definir i actualitzar l'esquema de la base de dades mitjançant anotacions a les classes del nostre model de dades. En aquestes anotacions sobre els atributs definim quin tipus de dades són i les relacions i cardinalitats amb les altres classes del nostre sistema.

A continuació es mostra un part de l'entitat *Experiment* per exemplificar el funcionament de l'ORM:

```

/**
 * @var \DateTime
 *
 * @ORM\Column(name="create_time", type="datetime")
 */
private $createTime;

/**
 * @var \DateTime
 *
 * @ORM\Column(name="update_time", type="datetime")
 */
private $updateTime;

/**
 * @var \DateTime
 *
 * @ORM\Column(name="lastDataInput_time", type="datetime", nullable=true)
 */
private $lastDataInputTime;

/**
 * @var boolean
 *
 * @ORM\Column(name="isOpen", type="boolean")
 */
private $isOpen = true;

/**
 * @ORM\ManyToOne(targetEntity="Laboratory", inversedBy="experiments")
 * @ORM\JoinColumn(name="laboratory_id", referencedColumnName="id")
 */
protected $laboratory;

/**
 * @ORM\OneToMany(targetEntity="Device", mappedBy="experiment")
 */
protected $devices;

```

*Il·lustració 72: definició de part de l'esquema E/R mitjançant atributs a la classe de l'entitat Experiment*

Com es pot observar a la figura anterior, els atributs tenen un comentari (*Annotation*) que serveix per indicar a l'ORM *Doctrine* que es vol tenir persistència d'aquella atribut del Model. L'ORM *Doctrine* s'encarregarà de mantenir l'esquema actualitzat (si es tracta d'una base de dades relacional) de la base de dades per tenir persistència d'aquell objecte amb els seus atributs.

Els atributs *\$laboratory* i *\$devices* permeten definir la relació de la classe amb les altres classes *Laboratory* i *Device* respectivament, fet que l'ORM tradueix en la corresponent relació de PK-FK i les seves restriccions d'integritat referencial a la base de dades.



## 7.5 Control de versions GIT

Al llarg de tot el desenvolupament del projecte s'ha utilitzat GIT com a eina de control de versions. S'ha creat un nou repositori GIT al servei *Bitbucket*<sup>28</sup> dins de l'equip de desenvolupadors de *CaminsTech*. Hem mantingut diferents versions del nostre projecte en branques.

- **Master:** branca principal del projecte; estable i certificada pel seu ús a l'entorn de producció.
- **Devel:** branca funcional on s'integren les funcionalitats noves abans de validar-les per a l'entorn de producció. Quan totes les noves funcionalitats han estat provades i correctament certificades, es realitza una operació de *merge* amb la branca de producció «*Master*».
- Branques individuals per cada funcionalitat: a cada nova implementació de funcionalitat que s'ha iniciat, s'ha realitzat una branca específica que ha servit per aïllar la branca «*Devel*» de possibles problemes d'estabilitat. Quan la funcionalitat ha estat finalitzada, aquesta s'ha fusionat mitjançant una operació de *merge* amb la branca «*Devel*».

Branch	Behind	Ahead
master MAIN BRANCH		
devel	3	90
ReadingRefactor	3	88
crear-maquina-vagrant	66	

Il·lustració 73: branques en un moment donat del repositori *CaminsSmartLab*

<sup>28</sup> <https://bitbucket.org/>

## 8. Planificació, anàlisi de costos i sostenibilitat finals

En aquest apartat es descriuen les modificacions que ha patit el projecte en la planificació i costos respecte d'aquelles inicials. Finalment, es fa un anàlisi de la sostenibilitat del projecte.

### 8.1 Modificacions realitzades a la planificació

A la [3.4 – Planificació temporal inicial](#) es va estimar un horari de 5 hores diàries, de 9:00 a 14:00 h, segons l'establert pel conveni de pràctiques amb l'Escola de Camins. Seguint aquesta planificació inicial, es preveia finalitzar tot el projecte com a data límit el 31 de desembre del 2015, finalitzant la implementació del sistema el 27 de novembre i preparant la memòria durant el mes de desembre, podent reservar una setmana addicional de desembre, per acabar de posar en marxa el sistema final.

Durant la realització del primer *Sprint* ja ens vàrem adonar de les dificultats de satisfer el calendari estipulat, a causa d'una corba d'aprenentatge més difícil de l'esperat amb algunes de les tecnologies emprades. Davant la impossibilitat d'ampliar les hores de dedicació diàries, els *sprints* originals han patit modificacions notables, havent d'afegir inclús un nou *Sprint* (*Sprint3bis*).

Davant les dificultats trobades i degut també a certa descoordinació entre els desenvolupadors i el *client*, els canvis a la planificació han estat els següents, indicat en color verd la dedicació real i en negre l'original.

Fases	Calendari	Durada (hores)	Durada (dies 5h/dia)
<b><i>Sprint Inception</i></b>	01/09/2015 – 25/09/2015	95 hores	19 dies
	01/09/2015 – 22/09/2015	80 hores	16 dies
<b><i>Sprint 1</i></b>	28/09/2015 – 16/10/2015	75 hores	15 dies
	23/09/2015 – 30/10/2015	140 hores	28 dies
<b><i>Sprint 2</i></b>	19/10/2015 – 06/11/2015	75 hores	15 dies
	02/11/2015 – 25/11/2015	90 hores	18 dies
<b><i>Sprint 3</i></b>	09/11/2015 – 27/11/2015	75 hores	15 dies
	26/11/2015 – 15/12/2015	70 hores	14 dies
<b><i>Sprint 3bis</i></b>	16/12/2015 – 31/12/2015	60 hores	12 dies
<b><i>Sprint 4</i></b>	30/11/2015 – 31/12/2015	120 hores	24 dies
	04/01/2016 – 27/01/2016	90 hores	18 dies
<b>Total</b>		440 hores	88 dies
		530 hores	106 dies

Taula 8: comparativa de costos de dedicació teòrics i reals

### 8.1.1 *Sprint Inception*: Investigació, planificació i posada en marxa de l'entorn de desenvolupament

Al llarg d'aquest *Sprint* es va dur a terme tot un procés d'investigació i planificació del projecte a més a més de la documentació de GEP que inclou l'estat de l'art i l'estudi de viabilitat econòmica. Aquest *Sprint* no va patir excessiva variació respecte de la seva planificació inicial (desviació negativa de -15 hores), malgrat que sí que ens va permetre preveure desviacions importants als futurs *Sprints* respecte de la seva planificació inicial.

Això ha estat degut a la corba d'aprenentatge més difícil de l'esperada amb *Symfony* i a l'estructuració d'un projecte d'aquesta envergadura. Vaig subestimar, erròniament, i en part per la descoordinació amb el *client*, el temps a invertir pel desenvolupament de totes les funcionalitats.

Durant aquest *Sprint* vaig començar a configurar l'entorn de desenvolupament, documentar-me sobre les eines que utilitzaria i el seu ús, i a definir els objectius del projecte. Vaig dissenyar els *mockups* i casos d'ús del sistema. Juntament amb això, es va realitzar la documentació de GEP, que malgrat servir-me d'ajuda per definir amb més rigor la planificació del projecte, no em va servir per evitar que subestimés el temps de dedicació real.

Finalment, aquest *Sprint* es va finalitzar 15 hores abans de l'estimat, previst pel 25/09/2015 i finalment assolint els objectius el 22/09/2015 amb un total de 80 hores de dedicació.

A continuació es mostren en detall els canvis temporals:

Nombre de tarea	Duración	Comienzo	Fin
▲ <b>Sprint Inception: Investigació, planificació i startup</b>	<b>19 días</b>	<b>mar 01/09/15</b>	<b>vie 25/09/15</b>
▲ <b>Investigació i contextualització</b>	<b>6 días</b>	<b>mar 01/09/15</b>	<b>mar 08/09/15</b>
Estat de l'art, context i abast del projecte	20 horas	mar 01/09/15	vie 04/09/15
Estudi de viabilitat econòmica	6 horas	lun 07/09/15	mar 08/09/15
▲ <b>Definició del projecte</b>	<b>8 días</b>	<b>mar 08/09/15</b>	<b>mié 16/09/15</b>
Planificació temporal	12 horas	mar 08/09/15	jue 10/09/15
Definició de tasques i full de ruta	22 horas	jue 10/09/15	mié 16/09/15
▲ <b>Eines de desenvolupament</b>	<b>8 días</b>	<b>jue 17/09/15</b>	<b>vie 25/09/15</b>
Prototipatge UX	15 horas	jue 17/09/15	lun 21/09/15
Aprenentatge eines desenvolupament	15 horas	mar 22/09/15	jue 24/09/15
Posada en marxa de l'entorn de treball	3 horas	jue 24/09/15	jue 24/09/15
Sprint Review	2 horas	vie 25/09/15	vie 25/09/15

Il·lustració 74: planificació temporal prevista inicialment per l'*Sprint Inception*

♣ <b>Sprint Inception: Investigació, planificació i startup</b>	<b>16 días</b>	<b>mar 01/09/15</b>	<b>mar 22/09/15</b>
♣ <b>Investigació i contextualització</b>	<b>4 días</b>	<b>mar 01/09/15</b>	<b>vie 04/09/15</b>
Estat de l'art, context i abast del projecte	15 horas	mar 01/09/15	jue 03/09/15
Estudi de viabilitat econòmica	4 horas	vie 04/09/15	vie 04/09/15
♣ <b>Definició del projecte</b>	<b>5 días</b>	<b>vie 04/09/15</b>	<b>jue 10/09/15</b>
Planificació temporal	8 horas	vie 04/09/15	lun 07/09/15
Definició de tasques i full de ruta	13 horas	mar 08/09/15	jue 10/09/15
♣ <b>Eines de desenvolupament</b>	<b>8 días</b>	<b>vie 11/09/15</b>	<b>mar 22/09/15</b>
Prototipatge UX	10 horas	vie 11/09/15	lun 14/09/15
Aprenentatge eines desenvolupament	21 horas	mar 15/09/15	lun 21/09/15
Posada en marxa de l'entorn de treball	7 horas	lun 21/09/15	mar 22/09/15
Sprint Review	2 horas	mar 22/09/15	mar 22/09/15

Il·lustració 75: planificació real obtinguda per l'Sprint Inception

8.1.2 *Sprint 1*: Implementació Gestió de Laboratoris, Experiments, Plantilles de Dispositiu, Dispositius i Lectures de Dispositiu

Al llarg d'aquest *Sprint* es va iniciar la implementació de la gestió de totes les entitats involucrades del nostre Model. Malgrat iniciar-se 5 dies abans del planificat, ràpidament ens vam adonar que la implementació seria bastant més costosa en temps del planificat inicialment degut al fet que ens adonéssim que hi havia certes funcionalitats repetitives que era millor reconsiderar la seva *refactorització*. Això ens va obligar a reimplementar certes funcionalitats d'una manera més òptima i modular mitjançant *Serveis*, fet que de retruc, ens va obligar a destinar més temps a investigar de nou.

A noves funcionalitats, també es van haver de refer els tests unitaris de validació, excedint considerablement el temps invertit en aquesta tasca respecte del previst.

Malgrat aquesta desviació important de temps inicial, en finalitzar l'*Sprint* podem afirmar que ha valgut la pena aquest sobre-cost ja que ha permès que el sistema sigui més modular i reaprofitable.

Aquest *Sprint* es va finalitzar 13 dies després de l'estimat, previst pel 16/10/2015 i finalment assolint els objectius el 30/10/2015 amb un total de 140 hores de dedicació.

A continuació es mostren en detall els canvis temporals:

♣ <b>Sprint 1: Implementació Gestió Experiments i Sensors</b>	<b>15 días</b>	<b>lun 28/09/15</b>	<b>vie 16/10/15</b>
Definició estructura web	11 horas	lun 28/09/15	mié 30/09/15
Implementació Casos Ús Experiments, Laboratoris i Sensors	52 horas	mié 30/09/15	mié 14/10/15
Testing	10 horas	mié 14/10/15	jue 15/10/15
Sprint review	2 horas	vie 16/10/15	vie 16/10/15

Il·lustració 76: planificació temporal prevista inicialment per l'Sprint 1

⚡ <b>Sprint 1: Implementació Gestió de Laboratoris, Experiments, Plantilles de Dispositiu, Dispositius i Lectures de Dispositiu</b>	28 días	mié 23/09/15	vie 30/10/15
Implementació inicial d'entitats i controladors	31 horas	mié 23/09/15	jue 01/10/15
Investigació sobre refactorització i Serveis	18 horas	jue 01/10/15	mar 06/10/15
Re-implementació amb Serveis	75 horas	mar 06/10/15	lun 26/10/15
Unit Testing	14 horas	mar 27/10/15	jue 29/10/15
Sprint review	2 horas	vie 30/10/15	vie 30/10/15

II·lustració 77: planificació real obtinguda per l'Sprint 1

#### 8.1.3 Sprint 2: Implementació Gestió d'Usuaris i REST API

En aquest *Sprint* es va dur a terme la implementació de la gestió dels usuaris del sistema i la creació de la REST API per a la interconnexió del servei web amb els dispositius *Arduino*. Aquest *Sprint* ha patit un increment de dedicació de 3 dies addicionals.

El motiu més important en aquesta desviació és el fet del canvi d'opinió en la implementació del motor de gestió d'usuaris emprat. Inicialment es va començar a utilitzar el *FOSUserBundle*, un *bundle* de tercers que proporciona tota la gestió d'usuaris i els mecanismes de seguretat. Un cop es tenia en funcionament, vam considerar que aquest *bundle* oferia massa funcionalitats i que afegia complexitat innecessària a la nostra aplicació; per aquest motiu es va decidir reimplementar el motor de gestió d'usuaris per un de nostre més simple i que s'ajustés millor a les característiques del nostre entorn.

Referent a la implementació de la REST API no ens hem trobat cap imprevist que ens hagi fet patir cap desviació addicional a l'Sprint.

Finalment, aquest *Sprint* es va finalitzar 13 dies després de l'estimat, degut en part pel retard acumulat de l'anterior *Sprint1*, previst pel 06/11/2015 i finalment assolint els objectius el 25/11/2015 amb un total de 90 hores de dedicació.

A continuació es mostren en detall els canvis temporals:

⚡ <b>Sprint 2: Implementació Gestió Usuaris i REST API</b>	15 días	lun 19/10/15	vie 06/11/15
Implementació CU Gestió Usuaris	15 horas	lun 19/10/15	mié 21/10/15
Implementació CU Rest API + MongoDB	45 horas	jue 22/10/15	mar 03/11/15
Testing	13 horas	mié 04/11/15	vie 06/11/15
Sprint review	2 horas	vie 06/11/15	vie 06/11/15

II·lustració 78: planificació temporal prevista inicialment per l'Sprint 2

♣ <b>Sprint 2: Implementació Gestió Usuaris i REST API</b>	<b>18 días</b>	<b>lun 02/11/15</b>	<b>mié 25/11/15</b>
Implementació CU Gestió Usuaris amb FOSUserBundle	10 horas	lun 02/11/15	mar 03/11/15
Re-implementació Gestió Usuaris	20 horas	mié 04/11/15	lun 09/11/15
Implementació CU Rest API + MongoDB	45 horas	mar 10/11/15	vie 20/11/15
Testing	13 horas	lun 23/11/15	mié 25/11/15
Sprint review	2 horas	mié 25/11/15	mié 25/11/15

II·lustració 79: planificació real obtinguda per l'Sprint 2

8.1.4 *Sprint 3*: Implementació dels Tests, client HTTP per *Arduino Yun* i configuració del dispositiu en xarxa

En aquest *Sprint* es va dur a terme la implementació de Tests funcionals que actuen sobre els Controladors de l'aplicació i permeten validar el correcte funcionament de totes les funcionalitats del sistema. A més a més, es va a dur a terme una investigació sobre el funcionament de l'*Arduino Yun* per adaptar-lo a la xarxa *sense fils* de l'escola i poder realitzar peticions a la REST API del sistema.

Aquest *Sprint* es va finalitzar 12 dies després del previst inicialment, degut principal pel retard acumulat als anteriors *Sprints*, previst pel 27/11/2015 i finalment assolint els objectius el 15/12/2015 amb un total de 70 hores de dedicació, 5 hores menys del previst inicialment.

A continuació es mostren en detall els canvis temporals:

♣ <b>Sprint 3: Implementació aplicació Gestió Experiments</b>	<b>15 días</b>	<b>lun 09/11/15</b>	<b>vie 27/11/15</b>
Definició estructura Arduino Yun	15 horas	lun 09/11/15	mié 11/11/15
Implementació Tests Funcionals	54 horas	jue 12/11/15	jue 26/11/15
Unit Testing	4 horas	jue 26/11/15	jue 26/11/15
Sprint review	2 horas	vie 27/11/15	vie 27/11/15

II·lustració 80: planificació temporal prevista inicialment per l'Sprint 3

♣ <b>Sprint 3: Implementació dels Tests, client HTTP per Arduino Yun i configuració del dispositiu en xarxa</b>	<b>14 días</b>	<b>jue 26/11/15</b>	<b>mar 15/12/15</b>
Investigació Arduino Yun	5 horas	jue 26/11/15	jue 26/11/15
Implementació Tests Funcionals	53 horas	vie 27/11/15	vie 11/12/15
Testing	10 horas	vie 11/12/15	lun 14/12/15
Sprint review	2 horas	mar 15/12/15	mar 15/12/15

II·lustració 81: planificació real obtinguda per l'Sprint 3

8.1.5 *Sprint 3bis*: Implementació dels Tests, solució de *bugs* i canvis visuals.

En aquest *Sprint* es va acabar la implementació dels Tests funcionals que actuen sobre els Controladors de l'aplicació, solucionar *bugs* descoberts a última hora gràcies als Tests funcionals i es van acabar de polir detalls estètics a la capa de presentació.

Aquest *Sprint* no estava previst a la planificació inicial però va tenir una durada de 12 dies assolint els objectius i donant per acabada la implementació el 31/12/2015 amb un total de 60 hores de dedicació.

A continuació es mostra en detall el diagrama temporal:

<b>▲ Sprint 3bis: Implementació dels Tests, solució de bugs i canvis visuals.</b>	<b>12 días</b>	<b>mié 16/12/15</b>	<b>jue 31/12/15</b>
Finalització de Tests Funcionals	25 horas	mié 16/12/15	mar 22/12/15
Solució de problemes descoberts	28 horas	mié 23/12/15	mié 30/12/15
Testing	5 horas	mié 30/12/15	mié 30/12/15
Sprint review	2 horas	jue 31/12/15	jue 31/12/15

Il·lustració 82: planificació real obtinguda per l'Sprint 3bis

#### 8.1.6 Sprint 4: Documentació i lectura finals

En aquest *Sprint* final s'ha acabat de desenvolupar tota la documentació de la memòria i la lectura finals. A més a més, també s'ha acabat de re-implementar certes funcionalitats dels controladors per fer-les més modulars. Aquesta tasca s'ha realitzat a estones lliures fora de l'horari laborable amb el consentiment del cap del projecte de *CaminsTech* i s'ha realitzat un seguiment mitjançant reunions per *Skype*. No obstant això, la tasca principal d'aquesta iteració ha estat acabar la documentació final de la memòria, així com la preparació de la lectura pel torn d'abril.

Aquest *Sprint* es va finalitzar 19 dies després de l'estimat, degut principal pel retard acumulat als anteriors *Sprints*, previst pel 31/12/2015 i finalment assolint els objectius el 27/01/2016 amb un total de 90 hores de dedicació.

A continuació es mostren en detall els canvis temporals:

<b>▲ Sprint 4: Documentació final</b>	<b>24 días</b>	<b>lun 30/11/15</b>	<b>jue 31/12/15</b>
Redacció de la memòria	60 horas	lun 30/11/15	mar 15/12/15
Preparació de la lectura	50 horas	mié 16/12/15	mar 29/12/15
Supervisió i correcció de la documentació	10 horas	mar 29/12/15	mié 30/12/15

Il·lustració 83: planificació temporal prevista inicialment per l'Sprint 4

<b>▲ Sprint 4: Documentació final</b>	<b>18 días</b>	<b>lun 04/01/16</b>	<b>mié 27/01/16</b>
Redacció de la memòria	60 horas	lun 04/01/16	mar 19/01/16
Supervisió i correcció de la documentació	30 horas	mié 20/01/16	mié 27/01/16

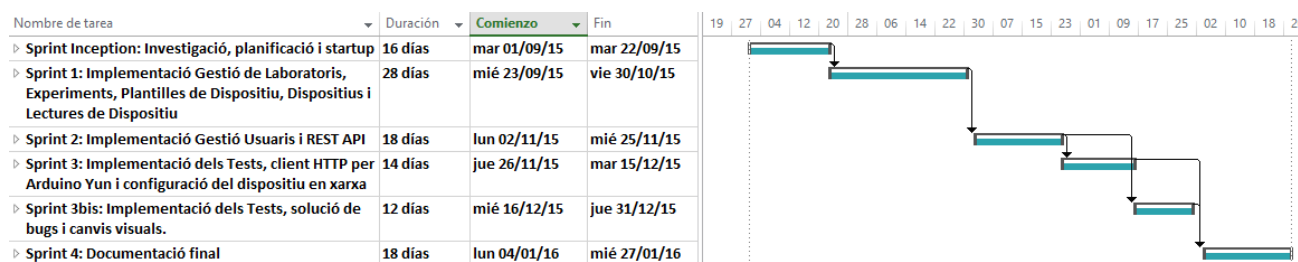
Il·lustració 84: planificació real obtinguda per l'Sprint 4



### 8.1.7 Diagrama de Gantt final compacte



Il·lustració 85: diagrama de Gantt inicial compacte



Il·lustració 86: diagrama de Gantt real compacte

A l'[Annex 2](#) es pot consultar el Diagrama de Gantt real complet i detallat.

## 8.2 Anàlisi econòmic final

Després d'haver finalitzat el projecte i d'haver detallat les modificacions que ha patit l'anàlisi inicial en el transcurs del projecte, s'ha pogut estimar el cost econòmic final de tot el projecte.

### 8.2.1 Costos directes per activitat

Les remuneracions dels recursos humans no han patit canvis respecte de l'anàlisi econòmic inicial.

Perfil	Remuneració anual (€)	Remuneració (€/h)
Cap de projecte	35.000 €	18,23 €/h
Analista funcional	33.000 €	17,19 €/h
Dissenyador UX	18.000 €	9,38 €/h
Programador Web	18.000 €	9,38 €/h
Programador Arduino/Python	18.000 €	9,38 €/h
Responsable QA	22.000 €	11,46 €/h

Taula 9: costos per perfil al final del projecte

A continuació es detallen els costos reals per cada una de les tasques del projecte amb assignació de perfil.

Activitat	Unitat	Perfil	Preu unitari (€/h)	Cost total (€)
Sprint inicial – Investigació i contextualització	19 h	Cap de projecte	18,23	346,37
Sprint inicial – Definició de requisits funcionals	8 h	Analista de requeriments	17,19	137,52



<b>Sprint inicial – Definició de tasques i full de ruta</b>	13 h	Cap de projecte	18,23	236,99
<b>Sprint inicial – Prototipatge UX</b>	10 h	Dissenyador UX	9,38	93,80
<b>Sprint inicial – Eines i entorn desenvolupament i Review</b>	30 h	Programador web	9,38	281,40
<b>Sprint 1 – Implementació inicial d'entitats i controladors</b>	31 h	Programador Web	9,38	290,78
<b>Sprint 1 – Investigació sobre refactorització i Serveis</b>	18 h	Programador Web	9,38	168,84
<b>Sprint 1 – Re-implementació amb Serveis</b>	75 h	Programador Web	9,38	703,50
<b>Sprint 1 – Unit Testing</b>	14 h	Software Tester	11,46	160,44
<b>Sprint 1 - Review</b>	2 h	Cap de projecte	18,23	36,46
<b>Sprint 2 – Implementació CU Gestió Usuaris amb FOSUserBundle</b>	10 h	Programador Web	9,38	93,80
<b>Sprint 2 – Re-implementació Gestió Usuaris</b>	20 h	Programador Web	9,38	187,6
<b>Sprint 2 – Implementació CU Rest API + MongoDB</b>	45 h	Programador Web	9,38	422,10
<b>Sprint 2 – Unit Testing</b>	13 h	Software Tester	11,46	148,98
<b>Sprint 2 – Review</b>	2 h	Cap de projecte	18,23	36,46
<b>Sprint 3 – Investigació Arduino Yun</b>	5 h	Programador Pt	9,38	46,90
<b>Sprint 3 – Implementació Tests Funcionals</b>	53 h	Programador Web	9,38	497,14
<b>Sprint 3 – Testing</b>	10 h	Software Tester	11,46	114,6
<b>Sprint 3 – Review</b>	2 h	Cap de projecte	18,23	36,46
<b>Sprint 3bis – Finalització de Tests Funcionals</b>	25 h	Programador Web	9,38	234,50
<b>Sprint 3bis – Solució de problemes descoberts</b>	28 h	Programador Web	9,38	262,64
<b>Sprint 3bis – Testing</b>	5 h	Software Tester	11,46	57,30
<b>Sprint 3bis – Review</b>	2 h	Cap de projecte	18,23	36,46
<b>Sprint 4 – Redacció de la memòria</b>	60 h	Cap de projecte	18,23	1093,80
<b>Sprint 4 – Supervisió i correcció de la documentació</b>	30 h	Cap de projecte	18,23	546,90
<b>Total</b>	<b>530 h</b>			<b>6271,74 €</b>

*Taula 10: costos directes per dedicació finals*

Concepte	Dedicació	Cost de mercat
Anàlisi econòmic inicial	440 h	5873,26 €
Anàlisi econòmic final	530 h	6271,74 €
Diferència	+ 90 h	+ 398,48 €

*Taula 11: comparativa de costos directes finals i inicials*

## 8.2.2 Costos indirectes

Els costos indirectes s'han incrementat a causa de l'increment de la durada del projecte. El preu unitari dels recursos, però, s'ha mantingut.

Producte	Unitats	Preu unitari	Dedicació	Cost estimat
Amortització maquinari	5 mesos	800 €/4 anys	90%	74,7 €
Amortització programari	5 mesos	17,75 €/mes	100%	88,75 €
Connexió dades	5 mesos	54 €/mes	50%	135 €
Transport	5 mesos	42,5 €/mes	100%	212,5 €
Impressió a paper	1000 pàgines	0,05 €/pàgina	100%	50 €
Total				560,95 €

*Taula 12: costos indirectes finals*

Concepte	Cost de mercat
Anàlisi econòmic inicial	458,94 €
Anàlisi econòmic final	560,95 €
Diferència	+ 102,01

*Taula 13: comparativa de costos indirectes finals i inicials*

### 8.2.3 Pressupost final

Refet el càlcul dels costos directes i indirectes actualitzat acord a la duració del projecte, obtenim que:

Pressupost final	Cost de mercat
<b>Costos directes</b>	6271,74 €
<b>Costos indirectes</b>	560,95 €
<b>Total</b>	<b>6832,69 €</b>

*Taula 14: pressupost final amb costos directes i indirectes*

El pressupost final s'ha vist alterat lleugerament. Aquest desviament s'ha minimitzat en part gràcies al pla de contingència i imprevistos que contemplava una possible desviació de 20 hores extra de dedicació, mentre que la desviació real ha sigut de 90 hores. No s'han contemplat imprevistos per avaries, ja que no han succeït finalment.

Pressupost inicial	Cost de mercat
<b>Costos directes</b>	5873,26 €
<b>Costos indirectes</b>	458,94 €
<b>Contingència</b>	1404,13 €
<b>Imprevistos</b>	155,04 €
<b>Total</b>	<b>7891,37 €</b>

*Taula 15: previsió de pressupost inicial*

D'aquesta manera, el cost total del projecte ha sigut de 6832,69 € respecte els 7891,37 € projectats inicialment (desviació negativa de 1058,68 €).

## 8.3 Sostenibilitat

### 8.3.1 Econòmica

El cost total del projecte final és de 6832,69 €, un preu que empreses grans podrien assumir però que la universitat pública, amb la difícil situació econòmica que viu avui dia, ho faria econòmicament inviable. No obstant això, malgrat que aquest projecte s'ha plantejat des del punt de vista empresarial, a la realitat aquest projecte serà desenvolupat per becaris de la UPC amb ànim d'investigar i aportar noves idees i eines a l'Escola de Camins. El cost real, per tant, és molt menor al pressupostat en aquest document, alhora que repercuteix en nous coneixements tant als alumnes com al personal docent i això té un valor incalculable.

Tot i així, es podria considerar l'ús de plataformes ja existents per implementar la solució, tot i que no acaben d'ajustar-se als requisits no funcionals del projecte, sí que són tècnicament solucions factibles i tenen un cost inferior al pressupostat. La viabilitat econòmica del projecte real és per tant de 8.

### 8.3.2 Social

L'objectiu principal d'aquest projecte és oferir una eina de centralització per experiments als laboratoris de l'Escola de Camins. Actualment ja existeixen eines privatives que realitzen aquesta missió per a equipament molt professional. Es volen realitzar experiments a escala més petita utilitzant plataformes lliures i més econòmiques sense dependre de solucions d'empreses externes que creïn una situació de dependència tecnològica.

En l'àmbit acadèmic, i en concret a la universitat pública, és de vital importància no deixar-se portar per les modes de mercat, sinó centrar-se en camps d'investigació que el personal docent cregui oportú. És per aquest motiu que es vol un sistema lliure i obert a tota la comunitat universitària interessada que serveixi per democratitzar els coneixements i resultats obtinguts als experiments que es realitzin a diferents assignatures d'aquesta escola.

El sistema estarà obert a tota la comunitat de la UPC i en l'idioma anglès per fer-lo accessible a tothom. La valoració en l'aspecte social del projecte és de 9.

### 8.3.3 Ambiental

Aquest projecte no ha de causar cap mal mediambiental greu. No obstant això, es genera certa despesa elèctrica per a l'espai de treball i pel transport que indirectament s'atribueixen a generació de CO2 però serà considerat mínim.

La infraestructura virtual no genera cap despesa elèctrica directa, ja que la infraestructura física on s'allotja aquesta està fora del nostre abast i no en tenim cap control. Un cop acabat el desenvolupament, el sistema passa a estar en màquines virtuals de la UPC.

Durant l'agost, per motius de manteniment i reducció de despeses, la UPC apaga tots els equipaments que no s'utilitzen en període docent i per tant es produeix una reducció energètica important respecte els servidors, però tampoc recauen en la responsabilitat d'aquest projecte.

És un projecte purament de software lliure, es podrà reutilitzar per a futurs projectes tant dins com fora de la UPC sense cap despesa extra en recursos materials. Té una puntuació de 9.

## 9. Millores pel futur

---

En aquest apartat es posen de manifest algunes possibles millores que s'han plantejat a l'equip de *CaminsTech* al llarg de tot el desenvolupament del projecte i que podrien aportar funcionalitats interessants als investigadors i persones implicades amb el projecte de cara a la realització d'experiments.

Una funcionalitat interessant seria la incorporació al sistema web d'un generador de gràfiques que permetés mostrar gràficament les dades obtingudes pels dispositius sobre els experiments. Aquesta gràfica hauria d'actualitzar-se automàticament en els casos d'experiments que s'estan executant actualment, fent-ho molt més atractiu per l'usuari envers haver de descarregar fitxers CSV i generar-se ell mateix les gràfiques amb alguna eina ofimàtica. Aquesta funcionalitat no és trivial, ja que cada experiment disposa d'un nombre de dispositius i sensors diferents amb diferents tipus de dades. La creació de la Plantilla de Dispositiu (*DeviceTemplate*) va orientat a facilitar la implementació d'aquesta funcionalitat en un futur.

Un altre punt a potenciar és la gestió dels permisos de compartició d'informació, permetent la compartició d'experiments i dispositius com si es tractés d'un sistema de fitxers, amb usuaris i grups, així com la possibilitat de fer públics experiments de cara a la compartició amb gent aliena al sistema.

Una funcionalitat interessant de cara al futur és la definició d'alarmes que permetin realitzar accions concretes, assolits uns valors de sensors especificats per l'usuari. Així, per exemple, es podria enviar un correu al responsable d'un experiment cada vegada que un sensor assoleixi un valor dins d'un rang definit per l'usuari; útil en experiments de llarga durada on l'investigador no pot estar consultant permanentment la informació. Aquestes regles serien programables per l'usuari responsable de l'experiment.

## 10. Conclusions

---

En finalitzar aquest projecte he pogut realitzar un procés de revisió de com s’han dut a terme totes les fases del projecte, prestant especial atenció a les decisions que s’han pres malament i quines alternatives es podrien haver aplicat per evitar-les. No obstant això, val a dir que els objectius principals del projecte s’han complert i el resultat ha estat satisfactori. També mencionar que ha estat un plaer poder fer un projecte real per a la Universitat Politècnica de Catalunya així com haver treballat amb l’equip de *CaminsTech*, agraïnt-los l’oportunitat de realitzar el meu treball final de grau amb ells.

### 10.1 Aprenent Symfony i l’arquitectura MVC

La introducció als *framework* web la vaig començar a l’assignatura ASW<sup>29</sup> de l’especialitat d’Enginyeria del Software. Aquest és el primer projecte desenvolupat per mi amb aquest *framework* i que té una utilitat al món real.

Malgrat la dificultosa *corba d’aprenentatge* que té aquest *framework*, val a dir que la documentació oficial és molt detallada i precisa, i m’ha permès relacionar molts conceptes apresos a l’especialitat de l’enginyeria del software amb els mètodes de programació que proposa *Symfony*.

*Symfony* suggereix una manera de programar que s’ajusta molt bé als patrons i metodologies recomanats al llarg de la carrera, cercant sobretot, la modularitat i el baix acoblament dels components per aprofitar al màxim la seva reutilització. A més, s’ha intentat programar seguint una filosofia de “*self-explained code*” per evitar haver d’escriure comentaris als blocs de codi de l’aplicació i facilitar la comprensió del codi a futurs programadors de l’aplicació.

Crec, per tant, que l’elecció d’aquest *framework* va ser encertada i garanteix a més un futur amb suport de l’aplicació.

### 10.2 Aprenent DevOps

Habituat a treballar en entorns *GNU/Linux* des de fa uns anys i a configurar manualment tots els serveis que requereix una aplicació web (servidor web, de base de dades, de PHP), el canvi en la manera de treballar gràcies a les eines de *DevOps* com *Vagrant*, ha resultat en un canvi molt positiu en l’aspecte TI del projecte.

Les eines *DevOps* mencionades ens han ajudat a tot l’equip de desenvolupadors a treballar d’una manera molt més àgil i segura en poder evitar problemes habituals en les configuracions dels servidors i delegar-ho a l’equip d’operacions.

---

<sup>29</sup> <http://www.fib.upc.edu/fib/estudiar-enginyeria-informatica/assignatures.html?assig=ASW>

Ha sigut molt positiu l'aprenentatge d'aquestes tecnologies i de ben segur que en faré ús a futurs projectes.

### 10.3 Aplicant la metodologia SCRUM

En aquest projecte he posat en pràctica la metodologia SCRUM. Hem intentat ser rigorosos amb el calendari i tècniques proposades per aquesta metodologia. No obstant això, és difícil acordar reunions setmanals amb totes les parts implicades per portar un seguiment.

Aquest motiu, juntament amb els imprevistos trobats i una subestimació dels temps de desenvolupament del sistema, és un dels principals motius pel desviament del temps respecte de la planificació inicial.

Aquest és un dels errors que m'han fet comprendre perquè sovint, molts projectes de software, s'acaben desviant respecte dels temps previstos inicialment.

No obstant això, estic satisfet amb les tècniques recomanades per aquesta referent a la gestió de tasques i seguiment continu per part de tot l'equip de treball.

## 11. Referències

---

- [1] «*Special Issue on Smart City and Internet of Things.*» <http://www.journals.elsevier.com/future-generation-computer-systems/call-for-papers/special-issue-on-smart-city-and-internet-of-things/> consultat el 20/07/2015.
- [2] «*Open source hardware design by GNU.*» <https://www.gnu.org/philosophy/free-hardware-designs.en.html> consultat el 20/07/2015.
- [3] «*Departaments, centres i laboratoris de recerca de l'Escola de Camins.*» <https://www.camins.upc.edu/recerca/departaments-centres-i-laboratoris-de-recerca> consultat el 21/09/2015.
- [4] «*Banda de radiofreqüència lliure ICM de baixa potència*» <http://www.boe.es/boe/dias/2015/04/10/pdfs/BOE-A-2015-3864.pdf> consultat el 21/09/2015.
- [5] «*How Will the Internet of Things Disrupt Your Performance Monitoring Strategy?*» <https://www.sevone.com/white-paper/how-will-internet-things-disrupt-your-performance-monitoring-strategy> consultat el 23/09/2015.
- [6] «*Which Programming Languages Can You Use With Arduino?*» <http://www.makeuseof.com/tag/programming-languages-can-use-arduino/> consultat el 23/09/2015.
- [7] «*SENTELO PERMITE ABARATIR COSTES A TRAVÉS DE LA COLABORACIÓN ENTRE CIUDADES.*» <http://smartcity.bcn.cat/es/sentilo.html> consultat el 18/06/2015.
- [8] «*Scrum Framework Alliance*» <https://www.scrumalliance.org/why-scrum> consultat el 28/09/2015.
- [9] «*MVC Framework – Introduction*» [http://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](http://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm) consultat el 19/10/2015.
- [10] «*ACID Properties*» [http://www.service-architecture.com/articles/database/acid\\_properties.html](http://www.service-architecture.com/articles/database/acid_properties.html) consultat el 23/10/2015.
- [11] «*Schema Design for Time Series Data in MongoDB*» <http://blog.mongodb.org/post/65517193370/schema-design-for-time-series-data-in-mongodb> consultat el 26/10/2015.



[12] «*Design patterns in Symfony2*» <http://www.slideshare.net/liuggio/design-pattern-symfony2> consultat el 10/12/2016.

[13] «*Web Vulnerabilities - PHP Security Cheat Sheet*» [https://www.owasp.org/index.php/PHP\\_Security\\_Cheat\\_Sheet](https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet) consultat el 25/12/2015.

## 12. Bibliografia

---

**La Guía Definitiva de Scrum: Las Reglas del Juego.** Consultat el 10/09/2015.  
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>.

**Recursos para la gestión de proyectos.** Consultat el 10/09/2015.  
<http://hipertextual.com/archivo/2013/02/gestion-de-proyectos-google-chrome/>

**Estudios de remuneración 2015.** Consultat el 10/09/2015.  
[http://www.pagepersonnel.es/productsApp\\_pp\\_es/Estudios%20Remuneracion/er\\_tecnologia.pdf](http://www.pagepersonnel.es/productsApp_pp_es/Estudios%20Remuneracion/er_tecnologia.pdf)

**Symfony 2.4, el libro oficial.** Consultat el 10/11/2014.  
[https://librosweb.es/libro/symfony\\_2\\_x/](https://librosweb.es/libro/symfony_2_x/)

**Object-Oriented Programming.** Consultat el 10/11/2014.  
<http://code.tutsplus.com/categories/object-oriented-programming>

**The Symfony Book.** Consultat el 10/11/2014.  
<http://symfony.com/doc/current/book/index.html>

**Getting Started with MongoDB.** Consultat el 09/09/2015.  
<https://docs.mongodb.org/getting-started/shell/>

**Serveis i recursos lingüístics. Idiomes a la UPC.** Consultat el 10/02/2016.  
<https://www.upc.edu/slt/ca>

## Glossari

---

**API:** (*Application Programming Interface*). Interfície *software* que especifica com diferents components informàtics poden interactuar entre ells.

**ARDUINO:** plataforma de *hardware* lliure basada en una placa amb un microcontrolador i un entorn de desenvolupament orientat a projectes multidisciplinaris.

**BACKEND:** capa de presentació d'una aplicació per a la gestió de les dades d'un sistema *software*.

**BUG:** error de programari (de l'anglès *bug*) resultat d'una fallada durant el procés de creació de programari.

**C++:** Llenguatge de programació dissenyat a mitjans dels anys 1980 per *Bjarne Stroustrup*.

**CAS:** (*Central Authentication Service*). Protocol de punt únic d'inici de sessió per la web.

**CLI:** (*Command Line Interface*) és una interfície que permet a l'usuari donar instruccions a un sistema per mitja de línies de text.

**CRAWLER:** Programa que inspecciona les pàgines del *World Wide Web* de forma metòdica i automatitzada.

**CSV:** tipus de document en format obert senzill per representar dades en forma de taula, en què les columnes se separen per punt i coma i les files per salts de línia.

**DOM:** (*Document Object Model*). Interfície de Programació d'Aplicacions (API) per a documents HTML i XML, una convenció multiplataforma i independent del llenguatge de programació que serveix per a representar documents HTML.

**FRAMEWORK:** Infraestructura de programari que, en la programació orientada a objectes, facilita la concepció de les aplicacions mitjançant la utilització de biblioteques de classes o generadors de programes.

**GIT:** *software* de control de versions pensat en l'eficiència i el manteniment de codi font.

**IOT:** (*Internet of Things*). Xarxa d'objectes de la vida quotidiana interconnectats mitjançant components electrònics i tecnologies sense fil.

**JSON:** estàndard obert basat en text dissenyat per a l'intercanvi de dades llegible per humans per a representar estructures de dades simples i llistes associatives.

**LAMP:** acrònim per descriure la plataforma sobre la que funcionen les aplicacions web creades utilitzant la combinació d'eines: *Linux*, *Apache*, *MySQL* i *PHP*.

**LGPL3:** tipus de llicència de *software* lliure que detalla com es pot distribuir lliurement un *software*.

**LOPD:** llei orgànica de protecció de dades de caràcter personal que té com a objectiu garantir i protegir les dades personals, les llibertats públiques i els drets fonamentals de les persones físiques, i especialment la seva intimitat i privadesa personal i familiar.

**LTS:** terme informàtic emprat per nombrar versions o edicions de software dissenyades per tenir suport durant un període més llarg del normal.

**ODM:** (*Object-document mapping*). Tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades no-relacionals basades en documents.

**OPEN DATA:** filosofia que busca que les dades estiguin obertes per tothom, sense restriccions, drets d'autor ni mecanismes de control.

**ORM:** (*Object-relational mapping*). Tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades relacionals.

**PROCESSING:** llenguatge de programació i entorn de desenvolupament integrat de codi lliure basat en Java per a la producció de projectes multimèdia creat per *Ben Fry*.

**PRODUCT BACKLOG:** llista de requisits funcionals ordenada per prioritats i definida per l'equip de desenvolupament en una metodologia SCRUM.

**PRODUCT OWNER:** a la metodologia SCRUM representa als *stakeholders* i és la veu del client.

**RASPBERRY PI:** ordinador monoplaca de baix cost desenvolupat en el Regne Unit per la Fundació *Raspberry Pi*.

**SAAS:** acrònim de *Software As A Service*. Model de distribució de software on el suport lògic i les dades són gestionades per una empresa externa i el client accedeix mitjançant Internet.

**SCRIPTS:** llenguatge de programació que controla aplicacions.

**SCRUM:** metodologia de desenvolupament de software àgil que es basa en processos iteratius i incrementals.

**SGBD:** (*Sistema Gestor de Base de Dades*). Conjunt de programes informàtics dissenyats per facilitar la gestió d'un conjunt de dades en una base de dades.

**SMART CITIES:** concepte de màrqueting en relació a les polítiques de desenvolupament basades en la sostenibilitat econòmica, social i ambiental.

**SPRINT:** iteració de la metodologia SCRUM i és la unitat bàsica de desenvolupament. La durada d'un *Sprint* és prefixada abans de l'inici d'aquesta i sol durar entre 1 setmana i 1 mes.

**SPRINT REVIEW:** en finalitzar un *Sprint*, l'equip es reuneix per avaluar l'estat de l'*Sprint* superat i definir el següent.

**STAKEHOLDERS:** grup de persones que tenen un interès concret en el suport d'una organització.

**STARTUP:** empresa emergent que busca arrancar, emprendre fent referència a noves idees de negoci, principalment suportades per la tecnologia.

**STATELESS:** protocol de comunicacions que tracta cada requesta com una transacció independent que no té relació amb cap altra requesta anterior.

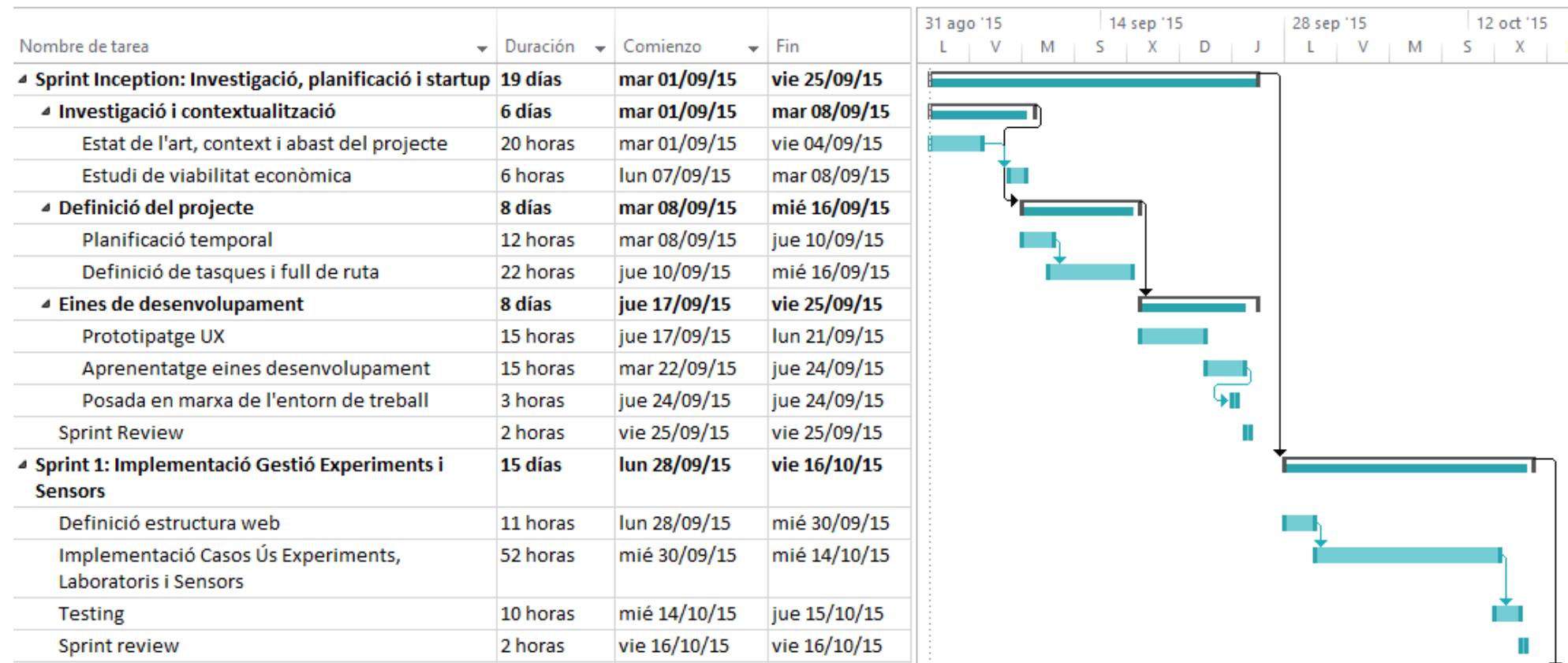
**STRING:** tipus d'estructura de dades que conté una seqüència de caràcters amb un ordre i una llargada determinades.

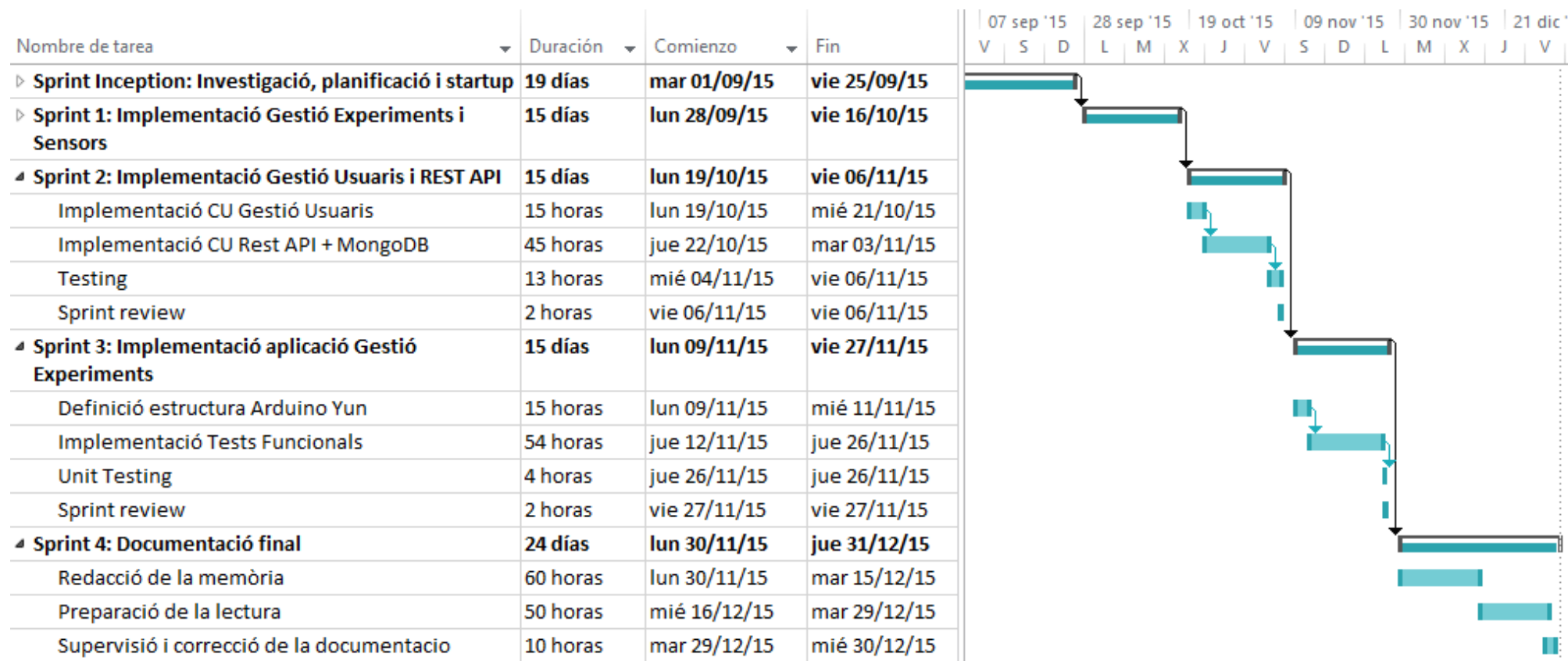
**TOKEN:** tipus de dada usat per les comunicacions en xarxa per identificar una sessió i l'intercanvi de missatges entre un client i servidor.

**UX:** experiència d'usuari, conjunt de factors en un entorn o dispositiu que genera una percepció positiva o negativa relatiu al disseny del hardware o software.

**W3C:** (*World Wide Web Consortium*). Consorci internacional dirigit per Tim-Verners-Lee, que treballa per a desenvolupar i promocionar estàndards per al *World Wide Web*.

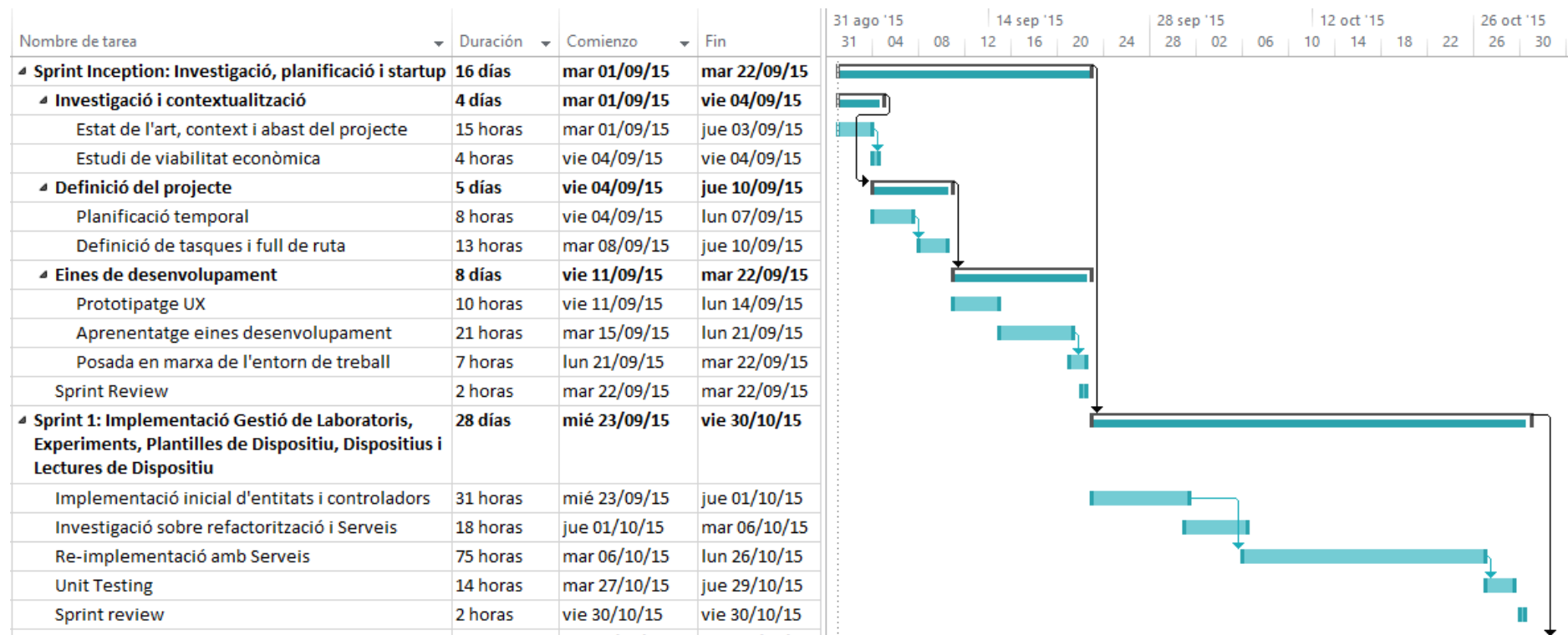
## Annex 1 – Diagrama de Gantt a la Planificació Inicial



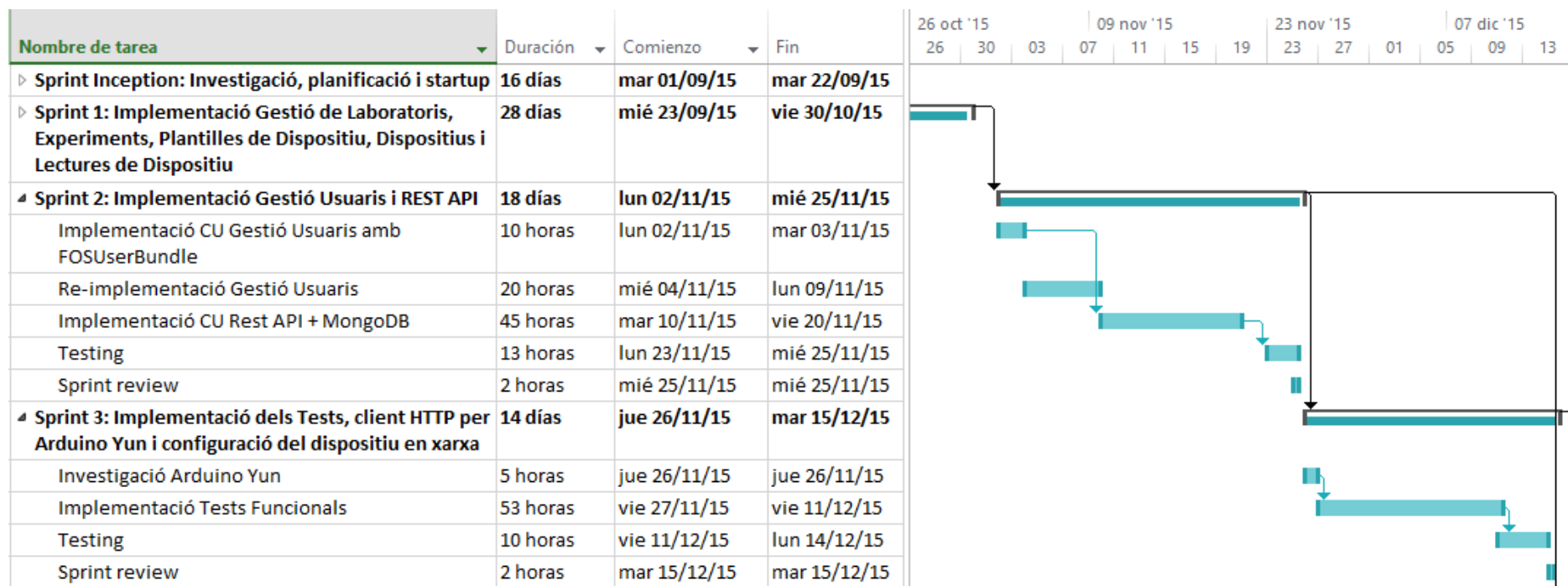


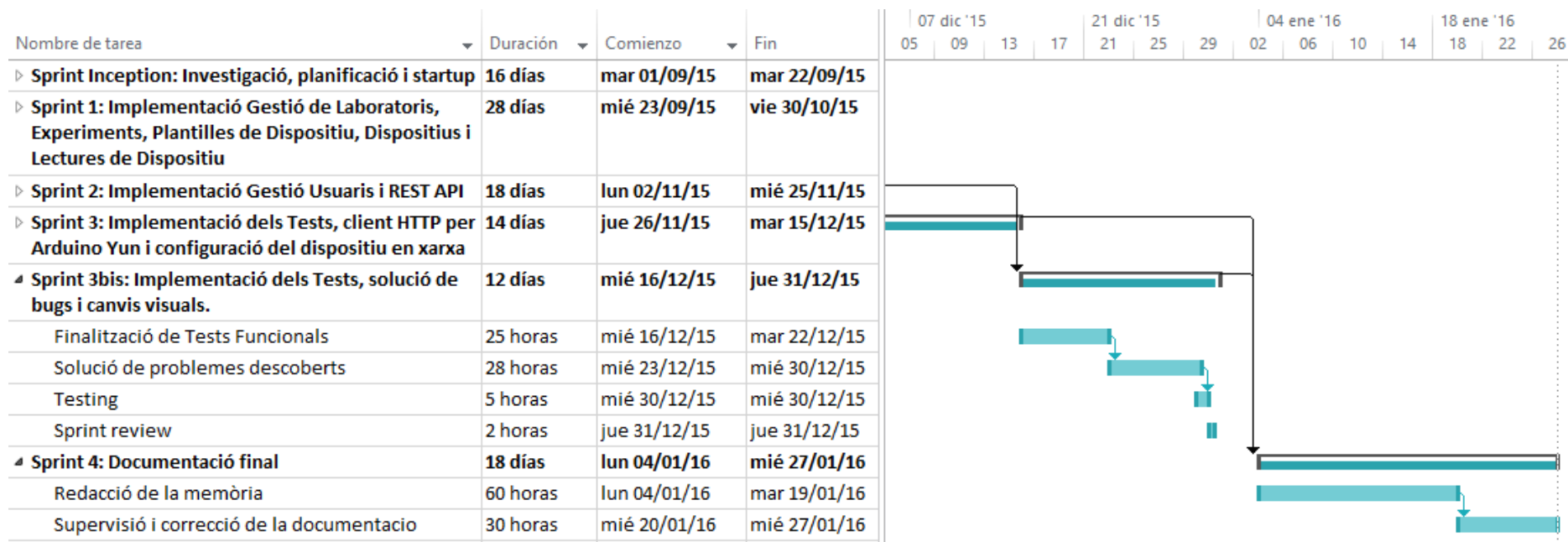
Annex 1: diagrama de Gantt complet per la Planificació Inicial

## Annex 2 – Diagrama de Gantt real









Annex 2: diagrama de Gantt amb la planificació real finalitzada

